Apple PowerKey: Complete Program Inside

COMPUTE!

$3.00
September
1986
Issue 76
Vol. 8, No. 9

$4.25 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

Beehive
An Irresistible Strategy Game For Commodore 64 And 128, IBM PC/PCjr, Atari, Apple, And Amiga

Dr. Sound For The 64
Easily Modify Sounds While Music Is Playing

A Full-featured ST Home Financial Calculator

Atari And 64 Speedup Techniques
Get Your Programs Off To A Fast Start

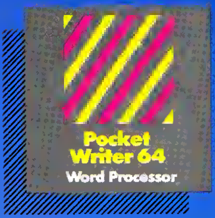
Easy IBM Animation

Amiga BASIC Style
Take Advantage Of This Power Language

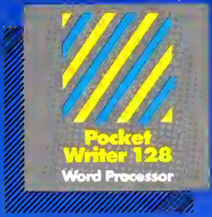
Enhancements For Atari And 64 *SpeedCalc* Spreadsheets

0 71486 02193 09

Solutions!



PW 128/64 Dictionary
also available at $14.95 (U.S.)

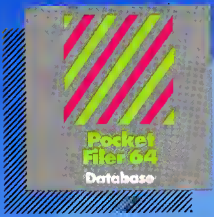


MAIL ORDERS:

CRYSTAL COMPUTER INC.
In Michigan 1·517·224·7667
Outside Michigan 1-800-245-7316

CANADIAN DEALER ENQUIRIES:

INGRAM CANADA LTD.
1-416-738-1700



Only The Name Is New

The professionol, full-feotured software line from Digitol Solutions is now called Pocket Software.
Pocket Writer 128/64.
Pocket Filer 128/64.
Pocket Planner 128/64.
The names are new, but this super software is still the same.

From now on, when you hear the word Pocket, it means software that's full-featured, handy and easy to use.

Pocket Softwore ot prices that won't pick your pocket.

Best-selling software for Your Commodore 128 or 64

You want the very best softwore you can find for your Commodore 128 or 64, right?

You wont integrated softwore — word processing, database and spreodsheet applications — at a sensible price. But, you also want top-of-the-line features. Well, our Pocket 128/64 software goes one better.

With Pocket 128 or 64, you'll find all the features you can imagine . . . ond then some. And Pocket 128/64 is so easy to use, you won't even need the reference guide. On-screen and in memory instructions will have you up and running in less than 30 minutes, even if you've never used a computer before.

The price? It's as low as you'd expect for o line of software colled 'Pocket'. Suggested Retail Price for the 64 software is $39.95 (U.S.) and $49.95 (U.S.) for the 128. Any of the 64 products moy be upgraded to their 128 version for $15.00 (U.S.) + $3.00 shipping ond handling. (Available to registered owners from Digital Solutions Inc. only.)

Pocket Writer 128 or 64, Pocket Planner 128 or 64 and Pocket Filer 128 or 64 . . . **Solutions** ot sensible prices from Digitol Solutions Inc.

Internotional & Distributor enquiries to:

Serious software that's simple to use.

Digital Solutions Inc.

30 Wertheim Court, Unit 2
Richmond Hill, Ontario
Canada L4B 1B9
telephone (416) 731-8775

Pocket Writer 128 ond 64 ore now ovoiloble in French.

Flight Simulator II
Scenery Disks

The Challenge of Accomplished Flight

With a realism comparable to (and in some ways even surpassing) $100,000 aircraft flight simulators, Flight Simulator II includes full flight instrumentation and avionics, and provides a full-color out-the-window view. Instruments are arranged in the format standard to modern aircraft. All the radios needed for IFR flight are included. Front, rear, left, right, and diagonal views let you look in any direction. Program features are clearly documented in a 96-page Pilot's Operating Handbook.

For training in proper flight techniques, Flight Simulator II includes another 96-page instruction manual, compiled by two professional flight instructors with over B,000 hours flight time and 12,000 hours of aviation teaching experience. You'll learn correct FAA-recommended flight procedures, from basic aircraft control through instrument approaches. To reward your accomplishments, the manual even includes a section on aerobatic maneuvers.

The Realism and Beauty of Flight

Go sight-seeing over detailed, realistic United States scenery. High-speed graphic drivers provide an animated out-the-window view in either day, dusk, or night flying modes.

Flight Simulator II features over 80 airports in four different scenery areas: New York, Chicago, Seattle, and Los Angeles. Six additional Scenery Disks covering the entire Western half of the United States are now available in IBM and C64/128 disk formats. Apple and Atari versions will be released soon. Each disk covers a geographical region of the country in detail, and is very reasonably priced.

The Pure Fun of "World War I Ace"

When you think you're ready, you can test your flying skills with the "World War I Ace" aerial battle game. This game sends you on a bombing run over heavily-defended enemy territory. Six enemy fighters will attempt to engage you in combat as soon as war is declared. Your aircraft can carry five bombs, and your machine guns are loaded with 100 rounds of ammunition.

See Your Dealer. Flight Simulator II is available on disk for the Apple II, Atari XL/XE, and Commodore 64/128 computers for $49.9S. Scenery Disks for the C64 and IBM PC (Jet or Microsoft Flight Simulator) are $19.9S each. A complete Western U.S. Scenery six-disk set is also available for $99.9S. For additional product or ordering information, call (800) 637-4983.

Apple II is a trademark of Apple Computer, Inc.
Atari XL and XE are trademarks of Atari Corp.
Commodore 64 and 128 are trademarks of Commodore Electronics Ltd.
IBM PC is a registered trademark of International Business Machines Corp.

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Order Line: (800) 637-4983
(except in Illinois Alaska, and Hawaii)

$10,000.00

Atari ST Programming Contest!

First Prize $5,000.00 **Second Prize $2,500.00**

Third Prize $1,000.00

Three Honorable Mentions $500.00 each

COMPUTE! Publications, Inc. is looking for the very best original software for the Atari ST series computers. And to prove we're serious, we're offering a total of $10,000.00 in prize money to the top six winners. That's $5,000.00 for First Prize, $2,500.00 for Second Prize, $1,000.00 for Third Prize, and $500.00 each for three Honorable Mentions. In addition, the winners will receive our standard royalties when their programs are published. And even if your program doesn't win a prize, you can still earn purchase fees and royalties if we accept your entry for publication.

Interested? If so, read these rules:

1. Entries must be your original work, previously unpublished. All those whose programs are accepted will be required to affirm this in writing.

2. You can submit as many entries as you want, but we cannot consider programs which have been entered in other contests or submitted for publication elsewhere at the same time.

3. The deadline is October 1, 1986. All entries must be received at our offices by this date. Programs submitted after this date will still be considered for publication, but will not be entered in the contest.

4. Entries are allowed (and encouraged) in virtually all software categories: home and business applications, education, recreation, telecommunications, graphics, sound and music, utilities, and desk accessories.

5. Entries may be written in any programming language—including BASIC, Logo, C, machine language, Pascal, Modula-2, Forth, FORTRAN, and Prolog—as long as they meet two requirements. First, if you're using a compiled language, the compiled object or run-time code must be a self-standing program that can be run by someone who doesn't own a copy of the language. (Exceptions are ST BASIC and Logo. Since these languages come with the ST, it can be assumed that everyone owns a copy.) Second, we must be able to legally distribute the program without incurring licensing fees or other obligations to the maker of the language. If you're not sure whether a certain language qualifies, contact its maker for clarification.

6. Entries must be submitted on a single- or double-sided 3½-inch ST disk with both the run-time code and source code included.

7. Entries must be accompanied by an article which explains how to use the program, what it does, and so on. If your program employs any new or unusual techniques that you think will be of interest to other ST programmers, you can also describe how the program works.

8. Submissions which do not win a prize and are not accepted for publication will be returned only if accompanied by a self-addressed, stamped mailer.

9. All judging will be handled by the staff of COMPUTE! Publications, Inc. All decisions regarding contest entries and acceptances will be solely at the discretion of COMPUTE! Publications, Inc., and all decisions are final. This includes decisions regarding creativity, similarity among entries, and so forth.

10. Winners will be announced by COMPUTE! Publications, Inc. in late 1986.

11. This contest is void where prohibited by law. Full-time, part-time & previous employees of COMPUTE! Publications, Inc., and Capital Cities/American Broadcasting Corporation are ineligible for the contest, but may still submit work for publication at standard rates.

Every Contest Entry Must Contain This Form:

I warrant that the program presently entitled____________________
______________________________ is my own original work and that the work has not been submitted for consideration elsewhere, nor has it been previously published. If my work is accepted by you, I understand that your decision as to the selection of winners and awarding of prizes is final and without recourse on my part. I agree, should you select my submission, to sign your standard contract, which includes assignment of the copyright of the program to COMPUTE!, and to allow you to use my name and image in promotional materials and other forms. *(If you are under age eighteen, your parent or legal guardian must sign for you.)*

Address entries to:
ATARI ST CONTEST
COMPUTE! Publications, Inc.
P.O. Box 5406
Greensboro, NC 27403

COMPUTE!™ Publications,Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies

Announcing major news for Atari ST users:

Special BONUS
First Issue FREE
with Subscription

COMPUTE!'s Atari ST DISK & MAGAZINE

A bimonthly magazine devoted exclusively to Atari ST enthusiasts that includes a disk containing all of the programs found in each issue.

Atari has proven the pessimists wrong. The Atari 520ST and 1040ST have become the bestsellers among the new generation of personal computers. Both are breakthroughs in price and performance, and the community of ST owners is growing by thousands each month.

That's one reason why COMPUTE! Publications is announcing a new magazine specially designed for ST users. At the same time, we recognize that the power of the ST presents a unique challenge to magazines which publish program listings. That's why we're including a 3½-inch disk that contains every program found in each issue—ready to load and run. No more typing!

Here's what you'll get in every issue of **COMPUTE!'s Atari ST Disk & Magazine:**

- **Top-quality programs.** Utilities. Games. Educational programs for youngsters. Application programs for home and business. And since all the programs will be on disk, there are few limitations on length or languages. A typical disk might contain an elaborate adventure game written in BASIC, a programming utility written in machine language, a dazzling graphics demo in compiled Pascal, and a useful home or business application written in Forth or C.

- **Neochrome of the Month.** Take a look at what computer artists are doing with the Atari ST. Each issue's disk contains a *Neochrome* picture file ready for you to load and admire. Are you an artist yourself? Send us a picture of your own, and we'll pay you if it's published.

- **Regular columns.** If you're a programmer—or would like to be—you'll love our columns on ST programming techniques and the C language. Or check out our column on the latest events and happenings throughout the ST community. Or send your questions and helpful hints to our Reader's Feedback column.

- **Reviews.** Honest evaluations of the latest software and hardware for the Atari ST.

- **News & Products.** A comprehensive listing of the newest releases for your ST.

- **And more:** Interviews with ST newsmakers, reports on the latest industry trade shows, and overviews of significant new product introductions.

Starting with the October issue (available September 1), **COMPUTE!'s Atari ST Disk & Magazine** will be found on newsstands nationwide for only $12.95 per copy, including disk. Or it can be delivered directly to your mailbox six times a year for only $59.95—a savings of over 20 percent.

As a special bonus, if you order a prepaid subscription before August 1, you'll get the first issue absolutely free!

To order, call 800-346-6767. In NY 212-887-8525
or send check or money order to
COMPUTE!'s Atari ST Disk & Magazine
ABC Consumer Magazines, Inc.
Circulation Dept./8th Floor
825 7th Avenue
New York, NY 10019

COMPUTE!™ Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies

Atari is a trademark of Atari Corporation

Introducing Certificate Maker.™

Because accomplishments deserve to be recognized.

Offer Congratulations! Say Thanks! Have Fun!

Giving someone a certificate is a wonderful way to recognize an outstanding achievement. It's also a perfect way to have a little fun.

Certificate Maker™ gives you **over 200 professionally designed certificates.** From strictly official to fun and witty, there's something for everyone and every occasion. So you can surprise a family member, praise a student, applaud an athlete and honor an employee with great looking certificates. And each one will be as personal, professional and special as you choose.

Personalized certificates in minutes.

Simply choose a certificate, select a border, type your message; add a date and signature… then print! It's that quick and that easy.

You can even create a name file and automatically personalize certificates for everyone in your class or club!

Over 200 exciting Certificates, Awards, Diplomas, and Licenses.

SPRINGBOARD™

Now available for Apple II+, IIe, IIc, and IBM PC • Springboard Software, Inc. • 7808 Creekridge Circle • Minneapolis, MN 55435 • (612) 944-3915

COMPUTE!

SEPTEMBER 1986
VOLUME 8
NUMBER 9
ISSUE 76

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 113 before typing in programs.

AP Apple, **Mac** Macintosh, **AT** Atari, **ST**, Atari ST, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **AM** Amiga. *General Interest.

COMPUTE! Publications,Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., 825 7th Ave., New York, NY 10019 USA. Phone: (212) 265-8360. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to: **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50950. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright ©1986 by COMPUTE! Publications, Inc. All rights reserved, ISSN 0194-357X.

Editor’s Notes

It’s always a pleasure here to launch a new magazine. It is something we pride ourselves on doing well, and our family of publications, both books and magazines, represents a significant and ongoing commitment to quality of product. We are a fluid group, at least internally, and have been fortunate in that we managed, as an editorial group, to avoid most of the pitfalls of overexpansion that befell many of our publishing colleagues in this industry’s jarring setbacks of 1984 and 1985. Tom Halfhill, most recently editor of COMPUTE!, has now taken the reins of our newest publication, COMPUTE’s Atari ST Disk & Magazine. It’s our most massive disk-based undertaking to date, and no publishing house in the history of this industry has ever dared place tens of thousands of bound-in disks into general newsstand distribution. Lance Elko, long our editor of COMPUTE’s GAZETTE, is expanding his duties to encompass COMPUTE!. We are confident this move will strengthen COMPUTE!, and help us in our continuing efforts to provide you with a constantly growing, and improving, publication. We welcome Lance to his new responsibilities, and can assure him, from long experience, that you out there will be the first to let him know how things are going.

A Software Product Note

While on the subject of COMPUTE!’s Atari ST Disk & Magazine, we’d like to mention an important concern. This is a truly integrated product—the magazine documents, nurtures, and tutors the disk. The programs, likewise, appear only on the disk. In short, you need the two parts to make the whole. One of our vendors’ biggest concerns for this magazine was that of removal of the disk. After all, they argued, this is an expensive item, and so on. It is of major concern to us that you, as potential readers, be able to handle the magazine and browse the printed pages. For this reason, you will find that the newest magazine we publish has a bound-in disk. And pages that open for previewing. We’re relying on you to prove us right. And, as always, COMPUTE! disk products are produced so that you can immediately, and easily, create your own backup. We do not engage in copy-protection. We expect you to refuse to engage in copying.

A Rare Exception

We do not frequently participate, in these pages, in a hand wringing regarding the ebbs and flows of our staff page. This is not, after all, afternoon television.

Our rare exception usually regards the move hither or yon of an editor or two as mentioned earlier in this piece. This month we must make a far more notable exception. Mr. Charles Brannon, of our resident staff, has accepted new employment, and we want not only to wish him well, but to devote to him a few sentences on this page. Charles, known by many of you as the author of *SpeedScript*, an incredibly sophisticated piece of COMPUTE!’s “giftware,” came to work for us in 1980 as a high school student, doing program listings after school. Over the years Charles grew and evolved into a very senior young member of our staff, achieving the position of program editor, and the person behind many, many of the significant programs we have developed and published here. We have many talented people, and would not wish these accolades for Charles to diminish that collective excellence. But there is, after all, only one *SpeedScript* and *Superfont*, and well, Charles, we’ll miss you, and we appreciate all the tremendous service you have provided to the readers and users of these publications over the last few years. We wish you well in your new venture.

Until next time, enjoy your issue. And watch for COMPUTE’s Atari ST Disk & Magazine, appearing on your local newsstand in early September.

Robert Lock

Robert C. Lock
Editor in Chief

USE THE BRAINS YOUR COMMODORE WASN'T BORN WITH.

Right at Your Fingertips in CompuServe's Commodore® Forums

Our Commodore Forums involve thousands of Commodore users worldwide. These Forums show you just how easy and fun it is to get the most from your Commodore Computer.

The Commodore **Communications Forum** provides the latest news on communications software and advice on how to effectively use your machine for online computing.

The Commodore **Programming Forum** supports programmers and developers of Commodore 8-bit computers.

The Commodore **Arts and Games Forum** is for all Commodore 8-bit computers. Compose or play music, create or retrieve colorful graphics, and download games software.

The Commodore **Amiga® Forum** is the national resource for all business and entertainment applications in the Amiga community.

Easy access to free software.

- *Download first-rate, non-commercial user-supported software and utility programs.*
- *Take advantage of CompuServe's inexpensive weeknight and weekend rates (when Forums are most active, and standard online charges are just 10¢ a minute).*
- *Go online in most major metropolitan areas with a local phone call.*
- *Receive a **$25.00 Introductory Usage Credit** when you purchase your CompuServe Subscription Kit.*

Information you simply can't find anywhere else.

Use the Forum *Message Board* to exchange mail with fellow members. Join ongoing, real-time discussions in a Forum *Conference*—with Commodore luminaries like Jim Butterfield, Jim Gracely, Bob Landwehr and Steve Punter. Scan Forum *Data Libraries* for free software, documentation and contributions from Commodore enthusiasts.

Enjoy other useful services, too. Like electronic editions of your favorite magazines, newsletters and articles, including Family Computing, OMNI Online and the Electronic Gamer.™

All you need is your Commodore computer and a modem...or almost any other personal computer.

To buy your Subscription Kit, see your nearest computer dealer. Suggested retail price is $39.95. To receive our free brochure, or to order direct, call 800-848-8199 (in Ohio, call 614-457-0802). If you're already a CompuServe subscriber, type GO CBMNET (the Commodore Users Network) at any ! prompt to see what you've been missing.

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, call 614-457-0802

An H&R Block Company

Publisher	James A. Casella
Founder/Editor in Chief	Robert C. Lock
Senior Editor	Richard Mansfield
Managing Editor	Kathleen Martinek
Executive Editor	Selby Bateman
Editor, COMPUTE!'s Atari ST Disk & Magazine	Tom R. Halfhill
Assistant Editor, COMPUTE!	Philip Nelson
Production Director	Tony Roberts
Production Editor	Gail Cowper
Editor, COMPUTE!'s GAZETTE	Lance Elko
Technical Editor	Ottis R. Cowper
Assistant Technical Editor	George Miller
Program Editor	Charles Brannon
Assistant Editor, COMPUTE!'s GAZETTE	Todd Heimarck
Assistant Features Editor	Kathy Yakal
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Victor, Kevin Mykytyn, Tim Midkiff
Copy Editor	Tammie Taylor
Submissions Reviewer	Mark Tuttle
Programming Assistants	David Florance, David Hensley
Executive Assistant	Debi Nash
Administrative Assistants	Julia Fleming, Iris Brooks, Mary Hunt, Sybil Agee
Associate Editors	Jim Butterfield Toronto, Canada Fred D'Ignazio Birmingham, AL David Thornburg Los Altos, CA
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Stephen Levy
Assistant Editors	Gregg Keizer, Ann Davies
Director, Book Sales & Marketing	Steve Voyatzis
Production Manager	Irma Swain
Art & Design Director	Janice R. Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Debbie Bray, Dabney Ketrow
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair
Director of Advertising Sales	Peter Johnsmeyer
Associate Advertising Director	Bernard J. Theobald, Jr.
Production Coordinator	Kathleen Hanlon
Promotion Assistant	Caroline Dark
Customer Service Manager	Diane Longo
Dealer Sales Supervisor	Orchid Tamayo
Individual Order Supervisor	Cassandra Green
Receptionist	Anita Armfield
Warehouse Manager	John Williams

James A. Casella, President
Richard J. Marino, Vice President, Advertising Sales
Christopher M. Savine, Director, Finance & Planning

COMPUTE! Publications, Inc. publishes:

COMPUTE!
The Journal For Progressive Computing

COMPUTE!'s GAZETTE
For VIC 20 And Commodore 64 Personal Computers

COMPUTE! Books

COMPUTE!'s GAZETTE DISK

COMPUTE!'s Apple Applications Special

Editorial offices:	324 West Wendover Avenue Suite 200 Greensboro, NC 27408 USA
Corporate offices:	825 7th Avenue New York, NY 10019 212-265-8360
Customer Service:	800-346-6767 (In NY 212-887-8525)
Hours:	9:30 A.M.–4:30 P.M. Monday–Friday

Coming In Future Issues

File Compressor And SpeedFile For The Commodore 64

TurboDisk For Apple DOS 3.3

Prisonball: A Captivating Game For Apple, Atari, And Commodore 64

Atari High-Speed String Sort

IBM PC/PCjr Mini Assembler

Subscription Orders

COMPUTE!
P.O. Box 10954
Des Moines, IA 50340

TOLL FREE
Subscription Order Line
800-247-5470
In IA 800-532-1272

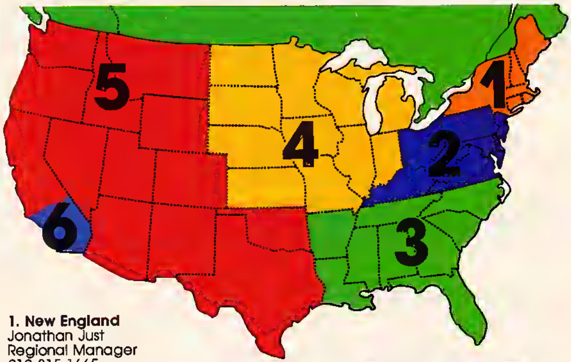
COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$24
	(two yrs.)	$45
	(three yrs.)	$65
Canada and Foreign Surface Mail		$30
Foreign Air Delivery		$65

Member
ABC
Audit Bureau of Circulations

MPA
Magazine Publishers Association

Advertising Sales



1. New England
Jonathan Just
Regional Manager
212-315-1665

2. Mid Atlantic
Jonathan Just
Regional Manager
212-315-1665

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
Dani Nunes
408-354-5553

6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales:
Peter Johnsmeyer

Associate Advertising Director:
Bernard J. Theobald, Jr.

COMPUTE! Home Office 212-887-8460.

Address all advertising materials to:
Kathleen Hanlon
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 10955, Des Moines, IA 50950. include a note indicating your preference to receive only your subscription.

Authors of manuscripts warrant that all materials submitted to COMPUTE! are original materials with full ownership rights resident in said authors. By submitting articles to COMPUTE!, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of COMPUTE! Publications, Inc. No portion of this magazine may be reproduced in any form without written permission from the publisher. Entire contents copyright © 1986, COMPUTE! Publications, Inc. Rights to programs developed and submitted by authors are explained in our author contract. Unsolicited materials not accepted for publication in COMPUTE! will be returned if author provides a self-addressed, stamped envelope. Programs (on tape or disk) must accompany each submission. Printed listings are optional, but helpful. Articles should be furnished as typed copy (upper- and lowercase, please) with double spacing. Each page of your article should bear the title of the article, date and name of the author. COMPUTE! assumes no liability for errors in articles or advertisements. Opinions expressed by authors are not necessarily those of COMPUTE!.

F-15 STRIKE EAGLE

For Fighter Pilots—*Not* Sightseers

Enjoyed by over **200,000** satisfied computer fighter pilots!

THE FIRST. THE BEST.
THE ONLY REAL COMBAT FLIGHT SIMULATOR.

MAJOR BILL
OVER
3000 FLYING HOURS

F-15 STRIKE EAGLE is the most realistic, accurate and fun combat flying simulation available. Others have tried to copy it. But none of them have ever strapped on a fighter to experience the thrill and excitement of real combat fighter missions like our Major Bill* has. Compare F-15 STRIKE EAGLE versus our famous competitors, better known for their fancy hangar graphics. If you want a **slow sightseeing hangar queen, buy theirs!** If you want a product that has had over 50 outstanding reviews, thrilled over two hundred thousand satisfied computer fighter pilots, and gives you **real** fighter pilot action, then you want F-15 STRIKE EAGLE!

COMPARE—

COMPARE	F-15 STRIKE EAGLE	IMITATION GAME
SPEED	4 TIMES FASTER	SLOW AND UNRESPONSIVE**
COMBAT AREA	7 REAL MISSIONS—LIBYA TO THE PERSIAN GULF	2 SMALL FICTITIOUS FLYING AREAS
COCKPIT	SOPHISTICATED NAVIGATION MAP MULTIPLE RADAR AND WEAPONS DISPLAY	TWO LINES OF TEXT
ACTION	MULTIPLE MISSION OBJECTIVES AND THREATS	PICK ONE MISSION OR ONE SET OF THREATS
VALUE	GREAT VALUE—(HUNDREDS OF HOURS OF PLAYTIME)	OVERPRICED

**C-64 COMPARISON

F-15 STRIKE EAGLE is for Commodore 64/128, Atari 800/XE/XL series, Apple II+ family, IBM PC and PC JR computers for a suggested retail price of only $34.95. Call or write for more information or Money Order, MC/Visa orders. Add $2.50 for shipping and handling.

Registered trademarks of Commodore Business Machines, Inc., Atari, Inc., Apple Computer Inc., and International Business Machines, Inc., respectively.

*Major Bill Stealey, USAF Reserve, President, MicroProse

MICRO PROSE
SIMULATION • SOFTWARE
120 Lakefront Drive • Hunt Valley, MD 21030
(301) 667-1151

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, *P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.*

STRING$, SPACE$, And CHR$

I have a suggestion for people who submit or translate IBM PC/PCjr programs for publication in your magazine. Whenever a BASIC program line requires that I type a long series of spaces, I find it difficult to tell exactly how many spaces are needed. This can be frustrating, because the "Automatic Proofreader" keeps signaling an error until I finally get the right number by trial and error. The STRING$ function can easily eliminate this problem. For instance, the statement PRINT STRING$(15,32) has exactly the same effect as PRINT "   " and is much easier to type in. STRING$ can be used where any long series of identical characters is needed. For instance, PRINT STRING$(40,46) prints a line consisting of 40 dots.

Richard J. Patton

This is an excellent suggestion, and the same general advice applies to every version of BASIC. Some versions include STRING$, which works exactly as in IBM BASIC; Amiga BASIC even includes a specialized SPACE$ function for creating a string of spaces. For BASICs that don't support either function, you can do the same job through concatenation. To create a string consisting of 30 spaces, for instance, use SP$=" ":FOR J=1 TO 30: SP$=SP$+CHR$(32): NEXT. This construction is easy to type and requires only a few more characters than printing the string in literal form.

For similar reasons, it's often preferable to express graphic characters or unusual symbols as CHR$ values rather than as string literals. Here are two different versions of a typical Commodore BASIC line:

```
10 IF X$="█" THEN GOSUB 100
10 IF X$=CHR$(133) THEN GOSUB 100
```

The first version of line 10 uses a literal graphics character to test whether the f1 function key has been pressed. The second version performs the same test with CHR$. To alleviate the "mysterious character" problem, our listing conventions (see "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue) replace any unusual Commodore or Atari character with a sequence that's easier to read. Here's what the same line would look like in a COMPUTE! listing:

```
10 IF X$="{ F1 }" THEN GOSUB 100
```

That's an improvement over listing an indecipherable graphics symbol, but it still requires that you remember the listing convention or look it up when the time comes. Of these three alternatives, the line with CHR$ is preferred in many cases, since it's easy to read and type, and doesn't require reference to anything but the listing. Of course, where large numbers of characters are involved, CHR$ may not be practical.

Spaced Out Operators

I enjoyed Bill Boegelein's "Amiga Puzzle" article in the May 1986 issue of COMPUTE!. I did have one problem, however, that may be of interest to your readers. The mistake was mine, not yours or the author's, but the solution might help everyone type in programs more accurately. The Play subroutine of Amiga Puzzle contains a complex IF statement that begins like this:

IF (mouseX>rat(x,y,0) AND ...

I mistakenly entered that portion of the statement like this:

IF (mouseX.rat(x,y,0) AND ...

Notice my inadvertent use of a period in place of the greater-than operator (>). Clearly, I forgot to hold down the SHIFT key when typing the > character. The problem arises because Amiga BASIC lets you include a period as part of a variable name. Instead of performing the logical comparison triggered by >, BASIC saw *mouseX.rat* as the name of an array. Of course, there is no such array or variable in the program, so its value was set to zero, like all other uninitialized variables. As a result, this part of the IF test is always false and the program's CheckCheat routine can never be called.

Although I was lucky enough to find this error without much searching, similar mistakes could be very difficult to detect in other situations. As a precautionary measure, I suggest that programmers always place a blank space on either side of a logical operator, as shown here:

IF (mouseX > rat(x,y,0) AND ...

If the original line had been written in this way, my typing error would have been much easier to spot. More to the point, BASIC itself would have detected the mistake and signaled a syntax error immediately. Again, the problem was mine, not Mr. Boegelein's or yours. But it could easily be prevented by following this simple rule.

Jack Purdum

Thanks for the suggestion.

SpeedScript File Resurrected?

I recently experienced an odd thing when using *SpeedScript* on my Commodore 128 in 64 mode. After writing a document, I pressed the RESET switch to go back to 128 mode. Then I decided to go back to 64 mode to finish up the document. When I reloaded and ran *SpeedScript*, I saw the same document that was in memory before I reset the computer. Shouldn't the memory have been cleared during this process? Does this mean that my 128 running in 64 mode isn't fully compatible with a normal 64?

Chris Hicks

To answer your last question first, this experience does not signal any sort of incompatibility. Your computer behaved exactly as a normal 64 with a RESET switch would under the same circumstances. The 64's reset routine does not *erase or scramble everything in the computer's memory; that happens only when you turn the computer off and on again. (For more details, see "64 RAM Report" in the June 1986 installment of this column.)*

SpeedScript *erases all of its text storage space when you first run the program, but not if you rerun it during the same session. When you run* SpeedScript, *it checks to see whether a special memory*

NEW!
- PAPERCLIP II for COMMODORE 128
- PAPERCLIP for APPLE II/E/C
- SPELLPACK for ATARI 130XE

Coming Soon
PAPERCLIP ELITE

THE Premier Word Processing Package

– CREATIVE COMPUTING

"PaperClip is easy to use, yet offers the advanced features of programs designed for the IBM PC. These include: block move, copy, delete, macros, automatic page numbering, headers, footers, underlining, boldface, super and subscripts, variable . character pitch, and custom character sets. The editing screen can be set up to 130 columns wide, & text can be scrolled in any direction. A preview mode displays formatted text exactly as it will appear on the printed page. You may further define your own formatting parameters, including margins, line lengths, page length and spacing.
PaperClip contains over 30 printer files for all the current major models. The documentation is excellent and the disk itself unprotected, though keyed through a joystick port. This means you can make as many back-up copies as you like, but can use the program only when the key is inserted." · CREATIVE COMPUTING

"PoperClip is one of the eosiest of the professionol word processors to use, with o sensible manual ond plenty of oids for the accident-prone." -COMPUTING NOW

"o "must hove" in on ideol software-library" -ELECTRONIC LEARNING

"PoperClip is the Codilloc of word processors" OMNI

"on excellent full-feotured word processor" THE BOOK OF CO

"So cleorly superior, ...Stote-of-the-ort word processing" · ANTIC *"...most*

"the ultimote word processor..." ANALOG

"...best professionol word processor ovoiloble" RUN

"exceptionol word processing" INPUT *"hord to beot"* ACE

"You'll find yourself growing spoiled." - FAMILY COMPUTING

'A superb word processor,...the most sophisticoted to dote!'

"...does exoctly whot it wos intended to do...ond more"..

"...most powerful of pockoges" -COMMODORE MAGAZINE

"...focts ottest to its excellence!" · FAMILY COMPUTING

"You will not find a word processing package superior to this one! CREATIVE COMPUTING

PaperClip

*"The #1 Best Selling Word Processing Package"**

-BILLBOARD'S COMPUTER SOFTWARE CHARTS!

Available for Commodore 64, Pet, all Atari home computers, and new enhanced 128K versions for Apple II/e/c, Atari 130XE and Commodore 128. **COMING SOON:** Paperclip Elite for IBM MS DOS, Amiga and Atari ST.

BATTERIES INCLUDED

30 Mural Street
Richmond Hill, Ontario
L4B 1B5 Canada
(416) 881·9941
Telex 06·986·266

"The Energized Software Company!"

WRITE TO US FOR FULL COLOR CATALOG of our products for Commodore, Atari, Macintosh, Apple and IBM systems.

17875 Sky Park North,
Suite P, Irvine, California
USA 92714
(416) 881·9816
Telex. 509·139

FOR TECHNICAL SUPPORT OR PRODUCT INFORMATION PLEASE PHONE (416) 881-9816. © 1986 BATTERIES INCLUDED. MACINTOSH APPLE, ATARI, COMMODORE AND IBM ARE REGISTERED TRADEMARKS RESPECTIVELY OF APPLE COMPUTERS INC., ATARI INC., COMMODORE BUSINESS MACHINES INC., AND IBM BUSINESS MACHINES INC. *AS COMPILED FROM NATIONAL RETAIL STORE SALES REPORTS FOR WEEK ENDING JANUARY 5, 1985. IF YOU CAN'T FIND PAPERCLIP AT YOUR FAVOURITE SOFTWARE OUTLET YOU CAN ORDER DIRECT FROM US AT THE FULL SUGGESTED LIST PRICE (PLUS $5.00 FOR POSTAGE AND SHIPPING) BY CALLING 1-800 387-5707. C64, ATARI & APPLE PAPERCLIP $59.95. PAPERCLIP WITH SPELLPACK FOR THE C64/128 and PAPERCLIP II FOR THE C128 $79.95. ALL PRICES SHOWN ARE IN U.S. DOLLARS *location contains the "I was here before" flag. If this flag is present,* SpeedScript *concludes that it was used previously in this session and sets up without erasing any text. Resetting the computer doesn't disturb either the memory area where text is stored or the location that holds the flag. So when you reran* SpeedScript, *the text was still there.*

This feature of SpeedScript *permits you to exit to BASIC if necessary, then reactivate the word processor without losing all of your work. As long as you don't load a different program or perform operations that change the contents of BASIC program space (or the memory location where* SpeedScript *stores the flag), any previous text should remain intact. To play it safe, of course, you shouldn't exit to BASIC more often than necessary.* SpeedScript *permits you to view the disk directory and send commands to the disk drive without leaving the program.*

1541 Disk Drive Rattle

I have seen a BASIC command that prevents the Commodore 1541 disk drive from knocking when protected software is loaded. Is there any way to prevent the knocking sound when you format a new disk? I am worried that too much knocking will force my drive out of alignment.

Tom Smith

While it's true that head-knocking isn't particularly good for the drive, there's no easy way to prevent it during the format process. The 1S41 drive is often called an "intelligent" peripheral because it contains its own microprocessor, free RAM, and operating system in ROM. The knocking sound heard when you format a disk is deliberate. It's caused by the format routine itself, which is permanently recorded in the drive's ROM.

A Commodore 1541 disk contains 35 tracks, numbered 1–3S. Track 35 is nearest the center hub, and track 1 is the outermost. The drive always begins formatting with track 1 and proceeds inward, formatting one track at a time. To locate the read/write head accurately for the beginning of this process, the drive steps the head outward *a total of 46 tracks. Since the drive is designed to access only 35 tracks in normal use, this maneuver is guaranteed to cause a read/write error regardless of the read/write head's initial position. The rattle is caused when the read/write head pounds against a mechanical metal stopper. The stopper physically prevents the head from moving past the outer edge of the disk.*

As you've seen, the command that prevents the head from knocking in other cases doesn't work when formatting. That method works by storing a smaller than usual number in location $6A in the drive's RAM. This location is a zero-page counter used to control how many times the drive should try to access a requested sector before giving up and signaling a read/write error.

The reason this trick doesn't work is that the ROM formatting routine, the relevant portion of which begins at $FAC7 in ROM, pays no attention to what's in location $6A. After stepping the head out 46 tracks, the ROM routine does set up a counter (at location $0620), but that's used to keep track of the number of errors encountered after *the head-knock takes place.*

It is possible to format a disk without rattling the head, but the alternatives are fairly involved and may be less reliable than the usual method. The first catch is that you need the ability to write a machine language routine for the drive to execute, download that code into one of the drive's RAM buffers, then cause the drive's microprocessor to execute it in place of the ROM format routine.

For those who are up to that challenge, here's one possibility: If your drive is correctly aligned, then, rather than locating the read/write head in the usual way, why not use a commercially formatted disk for calibration? Mass-produced commercial disks such as the 1541 Test/Demo disk are usually created on industrial equipment, not 1541 disk drives, and software companies have a strong incentive to keep such equipment in good alignment. So any commercial disk that doesn't contain deliberately implanted errors should be very close to the standard.

The idea is to insert the calibration disk, move the drive's read/write head to track 1 by reading track 1, sector 0, leave the read/write head stationary at that point, perform the other setup tasks required, then enter the ROM format routine at a point that bypasses the head-knocking section. That's a fairly tall order for most programmers and requires a much longer program than we can include in this space. This scheme could also increase the risk of inconsistent results, since it relies on two critical assumptions—that your drive is correctly aligned and that the calibration disk was accurately formatted in the first place—which may not be true in every case.

Loading Touch Tablet Screens In Atari BASIC

How can I write a BASIC program to display pictures drawn with the Touch Tablet and *Atari Artist* cartridge?

Peter Hinz

Loading Touch Tablet pictures in Atari BASIC is quite possible, and by calling an operating system routine, your BASIC program can load the images at machine language speed. But first, there are a few important points to cover.

To begin with, the Atari Artist *cartridge that comes with the Touch Tablet saves pictures in a special compacted format to conserve disk space. That's why, if you examine a disk directory of* Atari Artist *pictures, you'll notice that the files are usually of different lengths. Before you can load these pictures with a BASIC program, you have to convert them to uncompacted format.*

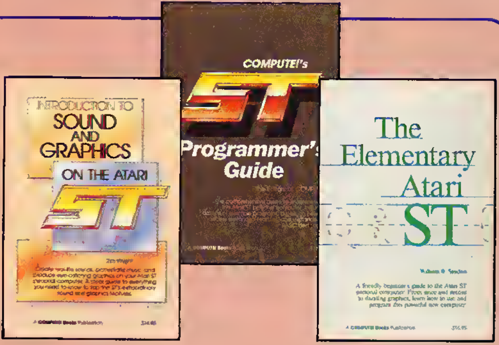
Although some people have written conversion utilities for this purpose, there's an even simpler method. It's not mentioned anywhere in the Atari Artist *manual, but if you hold down SHIFT and press the greater-than key (>),* Atari Artist *saves the current screen onto disk with the filename PICTURE. (Be aware that this replaces any existing file named PICTURE on the disk.) The file PICTURE is uncompacted and always takes up 62 disk sectors. This trick is useful in a couple of ways. It makes it possible to load* Atari Artist *pictures into other drawing programs for the Atari that use this format, including the Atari Light Pen's* Atari Graphics *cartridge and Datasoft's* Micropainter. *And it also makes it possible to load* Atari Artist *pictures into your own programs.*

But first, another point: Before loading the picture with a BASIC program, you have to set up the proper graphics mode. Atari Artist *(and most other drawing programs for the Atari) uses a special mode often known as GRAPHICS 7½. Of course, there's really no such thing as GRAPHICS 7½, but the term refers to the fact that this mode has the same horizontal resolution as GRAPHICS 7 (160 pixels) and the same vertical resolution as GRAPHICS 8 (192 pixels, without a text window). Yet, it also offers the same number of simultaneous screen colors as GRAPHICS 7 (four), while GRAPHICS 8 is limited to only two colors. Because it combines the best of both modes, GRAPHICS 7½ has been the most popular mode for drawing programs.*

GRAPHICS 7½ has always been supported by the Atari operating system. However, until the XL and XE series computers came out, it was not available from Atari BASIC without making some special POKEs to modify the display list. (The display list is an area of memory that tells the computer which graphics mode to display on the screen.) On an XL or XE, GRAPHICS 7½ is called GRAPHICS 15.

The following BASIC program shows how to load a 62-sector screen file named PICTURE at machine language speed. It should work with any uncompacted screen files, including those created with Atari Artist, *the Atari Light Pen, and* Micropainter. *This program is actually a slightly modified version of the program named MENU on the Atari* COMPUTE!

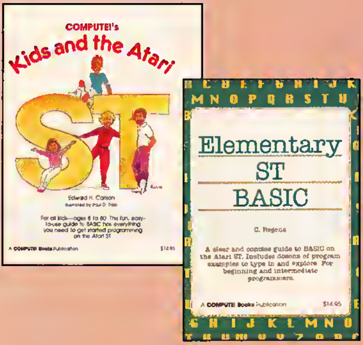
COMPUTE! Books'
ATARI ST Collection



INTRODUCTION TO SOUND AND GRAPHICS ON THE ATARI ST
COMPUTE!'s ST Programmer's Guide
The Elementary Atari ST

COMPUTE! Books offers you a superior line of titles for the new Atari ST. Packed full of useful utilities, exciting games, in-depth tutorials, and valuable applications, these clearly written books bring you fully tested information and entertainment for the whole family.

Look for these COMPUTE! books at your local book or computer store.



COMPUTE!'s First Book of the Atari ST

Edited
A valuable collection of ready-to-type-in-and-use applications, games, and utilities. Graphics utilities like "ST Doodler," games like "Switchbox" and "Tug-a-War," and educational programs like "Hickory Dickory Dock" turn your Atari ST into everything from a business graphics machine to a powerful teaching tool. Tutorials show you how to add power to ST BASIC and how to add excitement to your own creations with sound effects. *A disk is available for $15.95 which includes all the programs in the book, 203BDSK.*
(September release)
$16.95 ISBN 0-87455-020-3

The Elementary Atari ST

William B. Sanders, 272 pages
A friendly, easy-to-use guide to the Atari ST, this book takes you through connecting your computer, loading programs, creating graphics and music, and writing your own programs.
$16.95 ISBN 0-87455-024-6

Elementary ST BASIC

C. Regena, 208 pages
A tutorial and reference guide to the ST's impressive graphics, animation, and sound with complete descriptions of ST BASIC's commands, syntax, and organization. *A disk is also available for $15.95 which includes programs from the book, 343BDSK.*
$14.95 ISBN 0-87455-034-3

COMPUTE!'s Kids and the Atari ST

Edward H. Carlson, 238 pages
Easy-to-understand instructor notes, lessons, assignments, and lively illustrations help both kids and adults painlessly learn to program on the Atari ST. The latest in the bestselling series by this author.
$14.95 ISBN 0-87455-038-6

COMPUTE!'s ST Programmer's Guide

Editors of COMPUTE!, 356 pages
A comprehensive reference guide to the Atari ST, this book explores in detail Logo and BASIC, the advanced features of the ST such as GEM and TOS, and every aspect of programming from concepts to actual program writing.
$16.95 ISBN 0-87455-023-8

Introduction to Sound and Graphics on the Atari ST

Tim Knight, 197 pages
Thorough descriptions of the Atari ST's color graphics and sound abilities, plus all the information needed to create a complete sound and graphics system.
$14.95 ISBN 0-87455-035-1

You can order directly from COMPUTE! by calling 800-346-6767 (in NY call 212-887-8525) or by sending your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please include $2.00 per book shipping and handling for U.S. and surface mail or $5.00 for airmail. North Carolina residents add 4.5 percent sales tax.

Please allow 4–6 weeks for delivery from receipt of order.

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

TO ORDER CALL TOLL FREE

1-800-233-8950

DEPARTMENT A209

TELEX 5106017898

OR MAIL YOUR ORDER TO:

COMPUTER MAIL ORDER
Department A209
477 E. Third Street
Williamsport, PA 17701

POLICY

Add 3% (Minimum $7.00) shipping and handling. Larger shipments may require additional charges. Personal and company checks require 3 weeks to clear. For faster delivery use your credit card or send cashier's check or bank money order. Pennsylvania residents add 6% sales tax. All prices are subject to change and all items are subject to availability. Defective software will be replaced with the same item only. Hardware will be repaired or replaced at our discretion within the the terms and limits of the manufacturer's warranty. We cannot guarantee compatibility. All sales are final and returned shipments are subject to a restocking fee.

EDUCATIDNAL INSTITUTIONS CALL TOLL FREE

1-800-221-4283

CUSTOMER SERVICE & TECHNICAL SUPPORT
1-717-327-1450

CANAOIAN OROERS

1-800-268-3974
Ontario/Quebec

1-416-828-0866
In Toronto

1-800-268-4559
Other Provinces

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders. Call the Canadian Office for Canadian prices.

THE CMO ADVANTAGE

- Next day shipping on all in-stock items.
- Free easy access order inquiry.
- Orders from outside Pennsylvania save state sales tax.
- Free technical support from our factory trained technicians.
- There is no limit and no deposit on C.O.D. orders.
- There is no extra charge for using your Visa or MasterCard and your card is not charged until we ship.
- No waiting period for cashier's checks.
- We accept purchase orders from qualified corporations. Subject to approval.
- Educational discounts available to qualified institutions. (See the toll free educational phone number above.)
- FREE CATALOG MEMBERSHIP

HOME COMPUTERS

ATARI

65XE (64K)	**$89.99**
130XE (128K)	**$139.00**
520ST (S12K)	**$369.00**

520ST Monochrome System
- 520ST with modulator
- disk drive
- mouse
- logo
- Basic
- 1st Word
- monochrome monitor

LOW, LOW SYSTEM PRICE **$649⁰⁰**

520ST Color System
- 520ST with modulator
- disk drive
- mouse
- logo
- Basic
- 1st Word
- color monitor

LOW, LOW SYSTEM PRICE **$769⁰⁰**

800XL 64K	$69.99
1010 Recorder	$49.99
1050 Disk Drive	$149.00
1020 Printer	$29.99
1027 Letter Quality Printer	$129.00
1030 Direct Connect Modem	$59.99
Comrex 220 Atari	$89.99

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
IIc LCD Displey	$329.00

COMMODORE

Amiga Package
- 512K
- 2 - 3½" DS Disk Drives
- Amiga RGB Monitor.. **$1679.00**

C64 Package
- C64
- C1541
- Taxen 220 **$499.00**

C128 Package
- C128
- C1571
- NAP8562 Monitor **$779.00**

C128 Computer	**$269.00**
C1571 (Oisk Orive for C128)	**$248.00**
C1802 (RCB 13" Monitor for C128	**CALL**
C1670 (Modem for C128)	**$179.00**
C1530 Datasette	$39.99
C1660 Auto Modem	$59.99
Comrex 220 (C64 Interface)	$89.99
Xetec SuperGraphix 8K	$69.99
Micro R&D 128/64 Interface	$59.99
C128 Universal Monitor Cable	$19.99

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem 300/1200	$139.00
Signalman Express	$209.00
Lightning 2400 Baud	$329.00
Expressi (PC Haltcard)	$149.00
6480 (64/128) 300/1200 Beud	$139.00
520 1200 BPS (Atari ST)	$149.00
Lightening Half Card (PC)	$349.00
Omega 80 (Amiga)	$139 00

AST

Reach 1200 Baud Halt Card	$399.00

DICITAL DEVICES

AT300 - 300 Baud (Atari)	$99.99

EVEREX

1200 Baud Internel (IBM/PC)	$179.00

Hayes

Smartmodem 300	$139.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smertmodem 2400	$599.00
Micromodem IIe	$149.00
Smart Com II	$89.99
Chronograph	$199.00
Transet 1000	$309.00

Novation

Smert Cat Plus	$299.00
J-Cat	$99.99
Novation 2400	$499.00

QUADRAM

Quadmodem II	
300/1200	$339.00
300/1200/2400	$499.00

SUPRA

MPP-1064 AD/AA (C-64)	$69.99
1200 AT (Atari 8 Bit)	$149.00
1200 ST (520 300/1200 Baud)	$149.00

PORTABLE COMPUTERS

NEC

PC-8401 LS	$699.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Deta Recorder	$99.99

SPECIAL
- NEC 8201 Computer
- 8K Ram Cartridge
- 300 Baud Modem **$379⁰⁰**

SHARP

PC-1350	$149.00
PC-1261	$149.00
PC-1500A	$169.00
PC-1250A	$89.99
CE-125 Printer/Cessette	$129.00
CE-150 Color Printer Cassette	$149.00
CE-161 16K RAM	$129.00
PC-7000 PC Compatible	CALL

HEWLETT PACKARD

41CV	$139.00
41CX	$199.00
HP 11C	$49.99
HP 12C	$75.99
HP 15C	$75.99
HP 16C	$89 99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Cerd Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

We stock the full line of HP calculator products

ACCESSORIES

AMARAY

80 Column Printer Stand	$14.99

CURTIS

Side Mount SS-1	$19.99
Side Mount AT SS-2	$34.99
Universal Stand SS-3	$19.99
Diamond SP-1	$29.99
Emerald SP-2	$39.99
Sapphire SPF-1	$49.99
Ruby SPF-2	$59.99
Universal Printer Stand	$14.99
Static Mat	$29.99

DATA SHIELD

300 Watt Backup	$379.00
500 Watt Backup	$589.00
Turbo 350 Watt Backup	$449.00
P125 Power Director	$99.99
P150 Power Director w/Modem	$119.00

KENSINGTON

Master Piece	$99.99
Master Piece +	$119.00

KEYTRONICS

KB5150/KB5151/KB5151Jr.	CALL
KB5152B/KB5153/KB5149Jr.	CALL

MEMORY CHIPS

4164 RAM Chips	(ea.) $1.99
128 RAM Chips	(ea.) $12.99
256 RAM Chips	(ea.) $10.99

Polaroid

Palette	$1399.00
Power Processor	$229 00
Illuminated Slide Mounter	$39.99
Polacolor 2 Pack film	$18.99

DISKETTES

CENERIC

5¼" DS/DD	$10.99

IBM

5¼" DS/DD floppy disks (Box of 10)	$26 99

maxell

3½" SS/DD (10)	$18.99
3½" DS/DD (10)	$29.99
5¼" MD-1 SS/SD (10)	$11.99
5¼" MD-2 DS/DD (10)	$16.99
5¼" MD-2-HD for AT (10)	$29.99

SONY

MD1-5¼" SS/SD (10)	$9.99
MD2-5¼" DS/DD (10)	$13.99
MD2HD-5¼" High Density (10)	$28.99
MFD1-3½" SS/DD (10)	$18.99
MFD2-3½" DS/DD (10)	$29.99

Verbatim.

5¼" SS/DD	$12.99
5¼" DS/DD	$24.99
Disk Analyzer	$24.99

DISK HOLDERS

AMARAY

50 Disk Tub 5¼"	$9.99
30 Disk Tub 3½"	$9.99
100 Disk Tub 5¼" w/lock	$19.99

INNOVATIVE CONCEPTS

Flip'n File 10	$2.49
Flip'n File 50	$11.99
Flip'n File Data Case	$9.99

DRIVES

IOMECA

A110H Single 10	CALL
A210H 10 + 10	CALL
A120H Single 20	CALL
A220H 20 + 20	CALL
Save on 10 & 20 Carts	CALL

TALLCRAS5

25, 35, 50, 80 meg (PC)	trom $1299.00

IRWIN

Tape Backup	CALL

EVEREX

60 Meg Internal Backup System	$799.00
20 Meg Streamer	$699.00

CORE

AT20-AT72MB	CALL

MOUNTAIN

Hard Drives	CALL
Hard Card 20 MB	$749.00
Hard Card 30 MB	$999.00
Tape Back-Up	CALL

PRIAM

40, 60 MB Inner Space	CALL
Shared Data	CALL
Shared Space	CALL

RACORE

1500 Expansion Chessis	$379.00
1500 Chassis w/DMA	$429.00
2101 256K Memory	$179.00
2103 512K Memory	$199.00

CMS

10 Meg with controller	$369.00
20 Meg with controller	$499.00
Rambo-AT	$229.00

ALLIEO TECHNOLOCY

Apple II,II+,IIe ½ height	$109.00

INDUS

Atari GT	$189.00
C-64 /128 GT	$189.00

MSO

SD1 C-64 Single	$219.00
SD2 C-64 Dual	$469.00

TANOON

320K 5¼" (PC)	$119.00

TEAC

320K 5¼"	$119.00

DEPT. A209 CALL TOLL-FREE

COMPUTER MAIL ORDER

SOFTWARE FOR IBM

ANSA SOFTWARE	
Paradox	$499.00
ASNTON-TATE	
Framework II	CALL
dBase III Plus	CALL
BATTERIES INCLUOEO	
Isgur Portfolio	$159.00
BDRLANO	
Lightening	$59.99
Sidekick (unprotected)	$57.99
Rellex	$59.99
Travelling Sidekick	$44.99
Turbo Prolog	$64.99
CENTRAL POINT	
Copy II PC-Backup	$29.99
PC Qption Board	$84.99
DECISION RESDURCES	
Chartmaster	$229.00
Signmaster	$159.00
Diagram Master	$209.00
Map Master	$259.00
FIFTH CENERATION	
Fast Back	$99.99
FUNK SOFTWARE	
Sideways	$44.99
NARVARO SOFTWARE INC.	
Total Project Manager	$269.00
Presentation Graphics	$239.00
IMSI	
PC Mouse/Dr. Halo	$119.00
LIFETREE	
Volkswriter III	$149.00
LIVINC VIOEOTEXT	
Think Tank	$109.00
Ready	$64.99
LOTUS	
Symphony	CALL
1-2-3 Version 2	CALL
MECA SOFTWARE	
Managing Your Money 2.0	$99.99
Manage Your Market	$89.99
MICROPRO	
Easy	$94.99
WordStar 2000	$239.00
WordStar 2000 +	$289.00
WordStar Pro. w/GL Demo	$189.00
MICRORIM SOFTWARE	
R.Base 5000	$359.00
Clout 2.0	$129.00
MICROSOFT	
Flight Simulator	$34.99
MultiPlan	$129.00
Word 3.0	$279.00
Mouse	$139.00
MICROSTUF	
Crosstalk XVI	$89.99
Crosstalk Mark IV	$149.00
Remote	$89.99
MULTIMATE	
Multi Mate Word Proc	CALL
Advantage	$269.00
NOUNEMON	
Intuit	$69.99
NORTON	
Norton Utilities 3.1	$57.99
ONE STEP	
Golf's Best Pinehurst	$34.99
PFS: IBM	
Proof	$59.99
File/Graph	(ea.) $84.99
Report	$74.99
Write/Proof Combo	$84.99
PROFESSIONAL SOFTWARE	
Write-N-Spell	$89.99
TNE SOFTWARE CROUP	
Enable	$349.00
SATELLITE SYSTEMS	
Word Perfect 4.1	$219.00
SORCIM/IUS	
Accounting AP/AR/GL/INV/QE	(ea.) $299.00
SuperCalc III	$199.00
EasyWriter II System	$239.00
Super Project	$199.00
SUBLOCIC	
Jet	$37.99

PRINTERS

Canon	
A40,A50,A55	CALL
LBP-8A1 Laser	CALL
CITIZEN	
MSP-10 (80 col.)	$279.00
MSP-15 (132 col)	$419.00
MSP-20 (80 col.)	$349.00
MSP-25 (132 col.)	$509.00
C.ITOH	
Prowriter 1550P	$299.00
Starwriter 10-30	$299.00
3500 Tri Printer	$1499 00
corona	
Lazer LP-300	$2799.00
DIABLO	
D25 Daisywheel	$549.00
635 Daisywheel	$1099.00
D80IF Daisywheel	CALL
daisywriter	
2000	$699.00
EPSON	
Homewriter 10, LX-80	CALL
FX-85, FX-286, RX-100	CALL
DX-10, DX-20, DX-35	CALL
SQ-2000, Hi-80, HS-80, AP-80	CALL
LQ-800, LQ-1000	CALL
JUKI	
6000 Letter Quality	CALL
6100 Letter Quality	CALL
6200 Letter Quality.	CALL
6300 Letter Quality	CALL
6500 Letter Quality	CALL
5510 Dot Matrix Color	CALL
LEOEND	
808 Dot Matrix 100 cps	$179.00
1080 Dot Matrix 100 cps	$259.00
1380 Dot Matrix 130 cps	$289.00
1385 Dot Matrix 165 cps	$339.00
NEC	
3000 Series	$779.00
8000 Series.	$1099.00
ELF 360	$399.00
P560, P660, P760	CALL
OKIDATA	
182, 183, 192, 193, 2410, 84	CALL
Qkimate 10 (Specify C64/Atari)	$189.00
Qkimate 20(IBM), 282, 283	CALL
Qkidata 93	$299.00
Panasonic	
KX1080	$199.00
KX1091	$239.00
KX1092	$359.00
KX1592	$469.00
KX1595	$659.00
QUADRAM	
Quadjet	$399.00
Quad Laser	CALL
SILVER-REED	
500 Letter Quality	$189.00
550 Letter Quality	$299.00
800 Letter Quality	$699.00
star	
SG-10C (C64 Interface)	CALL
SB/SD/NX/SR Series	CALL
Powertype Letter Quality	CALL
Texas Instruments	
TI850	$529.00
TI855	$639.00
TI885	$799.00
TOSHIBA	
P321 (80 column)	$499.00
P341 (132 column)	$749.00
P351 (132 column)	$1049.00

MULTIFUNCTION CARDS

AST	
RamVantage	$349.00
Rampage-PC	$379.00
Six Pack Premium	$399.00
Six Pack Plus	$229.00
I/Q Plus II	$139.00
Advantage-AT	$369.00
Preview Mono	$299.00
PC Net Cards	$379.00
5251/11 Qn-line	$669.00
5251/12 Remote	$579.00
5251/11 Plus	$719.00
dca	
IRMA 3270	$879.00
IRMA Print	$999.00
IRMA Smart Alec	$779.00
EVEREX	
Edge Card.	$259.00
Graphics Edge	$219.00
Magic Card II	$159.00
Magic Card I	$99.99
EGA Video	$299.00
HERCULES	
Graphics	$299.00
Color	$159.00
IDEAssociates	
IDEA 5251	$549.00
INTEL	
PCNC8087 5MHz	
PCNC8087-2 8 MHz	
PCNC80287 6 MHz AT	CALL
PCNC802878 8 MHz AT	FOR
1010 PC-Above Board	YOUR
1110 PS-Above Board	PC
2010 AT-Above Board	
2110 AT-Above Board-PS	
MYLEX	
The Chairman	$439.00
PARADISE	
Color/Mono Card	$149.00
Multi Display Card	$199.00
High Res Mono	$169.00
Autoswitch EGA	$379.00
Modular Graphics Card	$269.00
PERSYST	
Bob Board	$359.00
QUADRAM	
Quadport-AT	$119.00
Liberty-AT (128K)	$349.00
The Gold Quadboard	$449.00
The Silver Quadboard	$239 00
Expanded Quadboard	$179 00
Liberty	$309 00
QuadSprint	$499.00
QuadLink	$399.00
QuadColor	$199.00
Quadboard-AT	$399.00
8600 E.G.A. card	$399 00
STB	
EGA Plus	$379.00
TECMAR	
Captain - 64	$199.00
Graphics Master	$469.00
VIOEO-7	
EGA	$399.00

INTERFACES

AST	
Multi I/Q (Apple II)	$149.00
DIGITAL DEVICES	
Ape Face (Atari)	$49.99
U-Print A (Atari)	$54.99
U-A16/Buffer (Atari)	$74.99
U-Call Interface (Atari)	$39.99
U-Print C (C64)	$49.99
P-16 Print Buffer	$74.99
U-Print 16 apple IIc	$89.99
MICRO R & D	
Apple IIc Parallel	$49.99
Kaypro 2000 Parallel	$49.99
C64/128	$59.99
Orange Mikro	
Grappler CD (C64)	$89.99
Grappler Plus (IIe, IIc)	$89.99
Grappter C (IIc)	$89.99
Grappler 16K (IIe, II+)	$139.00
PRACTICAL PERIPHERALS	
Graphicard	$69.99
Seriall Card	$99.99
Microbuffer II + 64K	$169.00
QUADRAM	
Microfazer	from $139.00
Efazer (Epson)	from $79.99

IBM

IBM PC SYSTEMS

Configured to your specifications.

Call for Best Price!

IBM-PC, IBM-XT, IBM-AT

AT&T	
Safari (7300)	CALL
6300	CALL
corona	
PPC400 Dual Portable	$1289.00
PPCXT 10 meg Portable	$1989.00
PC40022 Dual Desktop	$1389.00
PC400-HD2 10 meg	$1989.00
ITT	
ITT X-TRA	
256K, 2 Drive System	CALL
256K,10 meg Hard Drive System	CALL
XP5, 20 meg	CALL
KAYPRO	
KP-2000 Portable	CALL
Kaypro PC	CALL

PC TOO IBM CLONE

640K MB-256 installed, 150 Watt Power Supply, XT Keyboard, 360K Floppy, 20 Meg Hard Dnve......**$999.00**

SPERRY	
Sperry-AT	as low as $1749.00
Sperry-IT	as low as $2699.00

Call for Specific Configuration!

All Models	CALL

ZENITH

PC-138 Series, PC-148 Series, PC-158 Series, PC-160 Series, PC-171 Series, AT-200 Series......CALL

MONITORS

AMDEK	
Video 300A Amber	$129.00
Video 310A Amber TTL	$159.00
410 TTL Amber or Green	CALL
Color 600 Hi-Res. RGB	$399.00
Color 722 Dual Mode	$529.00
Color 725	CALL
Color 730	CALL
MACNAVOX	
515 RGB/Composite	$289.00
613 TTL Green	$99.99
623 TTL Amber	$99.99
NEC	
JB1280G TTL Green	$129.00
JB1285A TTL Amber	$129.00
JC1401 Multi Sync RGB	CALL
PRINCETON	
MAX-12 Amber	$179.00
HX-9 9" RGB	$469.00
HX-9E Enhanced	$519.00
HX-12 12" RGB	$469.00
HX-12E Enhanced	$559.00
SR-12 Hi-Res	$599.00
SR-12P Professional	$699.00
QUADRAM	
8400 Quadchrome I	$499.00
8420 Amberchrome	$179.00
8500 Quad Screen	$1449.00
TAXAN	
115 12" Green	$119.00
116 12" Amber	$129.00
121 TTL Green	$139.00
122 TTL Amber	$149.00
220 14" Color Composite	$179.00
620 640x200 RGB	$439.00
630 640x200 RGB	$489.00
640 720x400 RGB	$539 00
ZENITH	
ZVM 1220 Amber	$99.99
ZVM 1230 Green	$99.99
ZVM 1240 IBM Amber	$149.00
ZVM 135 RGB	$459.00
ZVM 1330 RGB	$459.00
ZVM 1360 RGB	CALL
ZVM 1380 E G Comp	CALL

1-800-233-8950 DEPT. A209

DISK. *It's easily adapted to your own BASIC programs. Briefly, here's how it works.*

Lines 10 and 160 create a very short machine language routine that is used later to call a high-speed loading routine in the operating system. Lines 170–177 set up graphics mode 7½ on any Atari computer. If your program is intended only for XL and XE models, you can replace these lines with a single statement such as 170 GRAPHICS 15+16. Line 190 opens the file PICTURE on disk and jumps to the subroutine at line 980. This subroutine, in turn, calls an operating system routine which loads the screen into memory at full speed. Line 200 simply loops endlessly so the picture stays on the screen. Press BREAK or SYSTEM RESET to end the program.

```
MC 1Ø OIM CID$(7)
AC 16Ø CIO$="hhh":CID$(4)=CH
       R$(17Ø):CIO$(5)="LV":
       CID$(7)=CHR$(228)
CO 17Ø GRAPHICS 8+16:OL=PEEK
       (56Ø)+256*PEEK(561)+4
AJ 172 SETCDLOR 4,Ø,12:SETCD
       LDR Ø,2,1Ø:SETCDLDR 1
       ,2,6:SETCDLDR 2,Ø,Ø
ND 175 PDKE DL-1,14+64:FDR I
       =2 TO 194:IF PEEK(OL+
       I)=15 THEN PDKE DL+I,
       14
BH 176 IF PEEK(DL+I)=15+64 T
       HEN PDKE DL+I,14+64
CH 177 NEXT I
II 19Ø DPEN #1,4,Ø,"O:PICTUR
       E":ADL=PEEK(88):ADH=P
       EEK(89):LN=7936:GDSUB
        98Ø:CLDSE #1
FN 2ØØ GDTO 2ØØ
EC 98Ø X=16:REM File#2,$2Ø
EF 99Ø ICCDM=834:ICBADR=836:
       ICBLEN=84Ø:ICSTAT=835
PL 1ØØØ PDKE ICBADR+X,ADL:PO
        KE ICBADR+X+1,ADH
LN 1Ø1Ø L=LN:H=INT(L/256):L=
        L-H*256:PDKE ICBLEN+
        X,L:POKE ICBLEN+X+1,
        H
PB 1Ø2Ø PDKE ICCDM+X,7:A=USR
        (ADR(CID$),X)
KI 1Ø25 RETURN
```

When the picture appears, chances are the screen colors won't be right. You'll have to recreate the picture's original colors with four SETCOLOR statements inserted somewhere between lines 170 and 190. You can figure out what these SETCOLOR statements should be by looking at the Color Menu screen in Atari Artist. *The four color register numbers along the bottom of the Color Menu screen—0, 1, 2, and 3—correspond to the first parameter in the SETCOLOR statement. Color 0 = SETCOLOR 4, color 1 = SETCOLOR 0, color 2 = SETCOLOR 1, and color 3 = SETCOLOR 2. The second parameter in SETCOLOR matches the color numbers along the vertical color bar on the Color Menu screen (0 to 15). And the third parameter in SETCOLOR is derived from the vertical luminance bar on the Color Menu screen (also 0 to 15, but use the even numbers only). For example, if color 0 in* Atari Artist *is set to black, your program would need a statement such as SETCOLOR 4,0,0.*

Incidentally, another undocumented trick makes it possible to load *uncompacted-format pictures into* Atari Artist, *too. Simply hold down SHIFT and press the less-than key (<). This way, you can take 62-sector pictures created with the Atari Light Pen,* Micropainter, *and other drawing programs and modify them with the Touch Tablet. If you then save this screen with* Atari Artist *in the usual way, it's converted to compacted format.*

Commodore SHIFT–SPACE

Sometimes when typing in programs from your magazine on my 64, I've come across a SHIFT–SPACE. When I press SHIFT and the space bar, it doesn't appear any different on my screen from the normal space. What does the SHIFT–SPACE character do?

Warren Frederick

There is a difference between the normal space character and shifted space. Although they appear the same on your screen, they are actually two separate ASCII characters. The normal space is CHR$(32) while the shifted space is CHR$(160). This distinction is probably not significant in every Commodore program where a {SHIFT–SPACE} appears. Many times, the programmer happens to be working in lowercase and types in an entire message with SHIFT LOCK down. When this happens, a shifted space appears in the listing, but an unshifted space would work just as well.

However, sometimes SHIFT–SPACE serves a special purpose. Certain programs use SHIFT–SPACE to mark a position on the screen that's invisible to the user. By PEEKing into screen memory, the program can distinguish between shifted and unshifted spaces even though both look identical on the screen.

You can also use SHIFT–SPACE to add short comments to disk filenames. If you include a shifted space as part of the filename, the disk drive treats that character as the end of the name and ignores any characters that come after it. But the extra characters are visible when you list the disk directory. For instance, you might want to save the current date to indicate when a program was last revised. This statement saves a program as FILE, followed by the date 9/22/86:

SAVE "FILE"+CHR$(160)+"/9/22/86",8

After you execute this statement, you can still load the program normally, with LOAD "FILE",8. But when you list the directory, the filename appears as FILE/9/22/86. This trick is frequently used when saving machine language programs, to indicate the SYS address used to start the program. Of course you are limited to a total of 16 characters, just as with any other disk filename.

IBM PrtSc Problems

When using the PrtSc function with my PCjr in "IBM Pie Chart Maker" (COMPUTE!, January 1985), my Gemini 10X prints the chart, but with thin blank lines between each row of the chart, as if the printer were displaying text lines. I have tried resetting the line space command to the printer and tested it in immediate mode to verify that the line space has been changed. But as soon as I type the PrtSc command, it seems that this command initializes the printer.

Rich Camaish

We've experienced the same problem when using PrtSc with anything except an Epson printer. Normally, pressing SHIFT–PrtSc just prints a text dump. In order to dump graphics with PrtSc, you need to enter the GRAPHICS command at the DOS command line to load the graphics print-screen driver. This driver was written specifically for the IBM Graphics Printer, a relabeled version of the Epson MX-80.

Apparently, the driver resets the printer completely before starting the graphics dump, as if the printer were turned off and on. (The Epson code for this is ESC-@.) It then sets the lines-per-inch to 8, corresponding to seamless eight-wire graphics printing. The code used for this function is different on the Gemini 10X and many other printers that are otherwise Epson compatible. Your printer accepts the reset sequence, though, throwing it back to nine lines per inch before starting the graphics dump. We've had the same problem with the IBM Color Printer.

The only way around this would be to modify the GRAPHICS driver. If you know something about 8088 machine language and have a working acquaintance with the DEBUG utility, you could search for the ESC-@ sequence (hex $1F $40) and replace it with two zeros to null it out. However, there are programs on the market and in the public domain that support graphics printing with PrtSc for many different printers. Check with your local IBM user group or nearest dealer to see if they've heard of these.

Apple HTAB In 80 Columns

I have an Apple IIe with an extended 80-column card. I found out recently that the Applesoft BASIC HTAB command does not work properly. When I type the following line in 80-column mode, I get an incorrect result:

```
HTAB 20:PRINT "THIS IS A TEST";:
  HTAB 1:PRINT "A"
```

SHELF-BASED EXPERT SYSTEM.

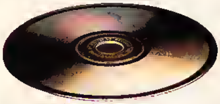
Chips for Health and Fitness
A Range of Chip Functions
An Overview of the System

An expert system is like borrowing someone else's experience and expertise on a given subject. UNDERSTANDING COMPUTERS by TIME-LIFE BOOKS is exactly that, an expert system on the subject of computers.

It's not software based, so it can't crash. And since it's not machine-specific, it won't become outdated easily, which means you'll find it an invaluable, long-lived reference whether you have an IBM®PC, an Apple IIe®, a multimillion dollar Cray®, or no computer at all right now.

A BRAND NEW SERIES THAT PRESENTS COMPUTERS IN A WHOLE NEW WAY.

UNDERSTANDING COMPUTERS is a new series of books that presents computers in a unique, broad-based way, unlike any other computer book you've ever seen. It gets "under the hood," right down to the nuts and bolts of computers to explain what you need to know about them in plain English.



On a digital record, sound is carried by pits and spaces, which are read by a laser beam.

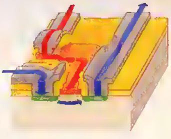
The first volume, *Computer Basics*, unwraps the mystery of writing binary code... the concept of logic gates... how chips are designed... how a light pen works for graphic effects... the sequence of events in a computer from the first clock pulse to the last. And much, much more.

Succeeding volumes cover *Software. Input/Output. Graphics. Networking.* The entire gamut of computer topics.

EVERY IMPORTANT CONCEPT IS ILLUSTRATED TO HELP ASSURE YOU UNDERSTAND.

Not only does UNDERSTANDING COMPUTERS give you all the computer background you want that you won't find elsewhere, it also shows you what it's all about. Full-color illustrations help make every concept crystal clear. And glossaries of terms help ensure your understanding. Each volume is a big 9¼" x 11⅛" hardcover book with approximately 128 pages.

FREE 10 DAY EXAMINATION. NO OBLIGATION.



How a transistor operates as a switch.

Begin to expand your computer know-how now by examining *Computer Basics* for 10 days free. Keep it and pay only $12.95 ($15.95 in Canada), plus shipping and handling. Then you can continue to receive other volumes every other month, always with a 10-day free trial. Keep only the books you want. And you can cancel any time. Or return *Computer Basics* and owe nothing.

Return the coupon today, and start to bridge the gap from computer buff to computer expert.

IBM is a registered trademark of International Business Machines Corporation. Apple IIe is a registered trademark of Apple Computer, Inc. Cray is a registered trademark of Cray Research, Inc.
© 1986. Time-Life Books Inc. Time & Life Building, Chicago, IL 60672-2058

UNDERSTANDING COMPUTERS STARTS HERE.

Mail to TIME-LIFE BOOKS, Inc.
Time & Life Building,
Chicago, IL 60672-2058

YES! Send me *Computer Basics*, as my introduction to the shelf-based expert system, UNDERSTANDING COMPUTERS. I agree to the terms outlined in this ad.

D1A3Y9

Name____________________

Address____________________

City____________________

State or Province__________ Zip or Postal Code__________

All orders subject to approval. Price subject to change

TIME LIFE BOOKS

UNDERSTANDING COMPUTERS

The computer prints this line preceded by 19 spaces:

THIS IS A TEST.A

Memory location 36 is supposed to contain the horizontal cursor position, but in 80-column mode, it always contains 0. The BASIC function POS(0) doesn't work either. How can I determine the current cursor position?

William Liao

Many older Apple II programs, especially those written in machine language, print to the screen by adding the horizontal cursor position (CH, location 36) to the address of the first character in the current row (BASL and BASH, locations 40 and 41), then storing a character at the address that results. When 80-column hardware is in use, this technique could scramble the Apple's memory, since the organization of 80-column screen memory is different.

As a precaution, whenever the Apple's I/O software accesses the 80-column screen to move the cursor or print, it resets CH to 0. This is why PEEK(36) and POS(0) no longer work. In IIe and IIc computers, the 80-column cursor position is kept in location 1403, called OURCH. (If you're familiar with the Apple II's memory arrangement, you'll remember that addresses between 1024 and 2047 are reserved for screen display memory. Since the 40-column screen is 40 × 24, that's a total of 960 bytes that are actually used. The 64 unused bytes are called screen holes and are used to store I/O variables. OURCH is one of these.)

The HTAB command changes the cursor's position by storing a new value in location 36. To keep this command operational, the enhanced I/O routines keep a copy of CH in another screen hole, location 1147 (OLDCH). Before each screen access, CH and OLDCH are compared. If they are different, CH must have been changed, so its value is made the current position by storing it in OURCH. The only time this doesn't work is when 80-column mode is active. Since CH and OLDCH are both set to zero at each screen access, an HTAB 1 command stores zero in CH, and there's no way to tell that anything happened. Since CH and OLDCH still contain the same value, OURCH is not altered.

One simple way to move the current screen position to the first column is to use a lone PRINT statement. All it does is move the cursor to the first column of the next line without disturbing the display at all. Another way to be certain of the cursor's position in any display mode is to POKE the new column value (0–79) into both CH and OURCH. In standard display mode (40 columns, checkerboard cursor), OURCH is not used; POKEing a value there doesn't seem to have any undesirable side effects.

When the enhanced I/O firmware is active (block cursor in 40 or 80 columns), you can find the current cursor column with PEEK(1403). To find the current column regardless of display mode, PEEK the value in CH. Then, if it has a value of zero, PEEK at 1403. This should always give the correct position.

EduCalc Clarification

A statement concerning disk initialization in the review of Grolier's *EduCalc* spreadsheet (March 1986) requires clarification. When using an uninitialized data disk, the program will automatically ask if you wish to initialize the disk and then lead you through an initialization routine. When using a disk that's already initialized, *EduCalc* recognizes that and skips the routine.

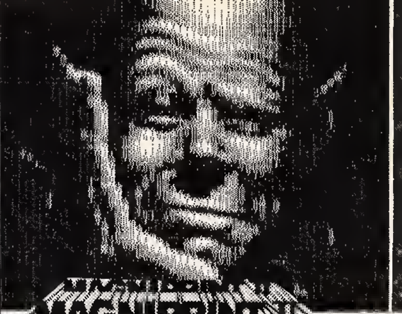
©

HOTWARE: Software Best Sellers

					Systems				
This Month	**Last Month**	**Title**	**Publisher**	**Remarks**	**Apple**	**Atari**	**Commodore**	**IBM**	**Macintosh**
Entertainment									
1.		*Elite*	Firebird Licensees, Inc.	Action/adventure	●		●		
2.	4.	*The Bard's Tale*	Electronic Arts	Adventure/role-playing game	●		●		
3.	2.	*Ultima IV*	Origin Systems, Inc.	Fantasy game	●	●	●		
4.		*Hardball*	Accolade	Baseball game	●		●		
5.		*Karate Champ*	Data East	Martial arts game	●		●		
Education									
1.		*Homework Helper: Math Word Problems*	Spinnaker	Math tutorial, high school level	●		●		
2.	1.	*Math Blaster!*	Davidson	Introductory math program, ages 6-12	●	●	●	●	
3.		*Spanish*	American Educational Computer	Spanish vocabulary skills	●	●	●	●	
4.	2.	*Music Construction Set*	Electronic Arts	Music composition program	●	●	●		
5.	3.	*Color Me:The Computer Coloring Kit*	Mindscape	Children's artistic tool	●		●		
Home Management									
1.	3.	*Better Working Spreadsheet*	Spinnaker	Spreadsheet	●		●		
2.	4.	*The Newsroom*	Springboard	Do-it-yourself newspaper	●		●	●	●
3.	1.	*Print Shop*	Brøderbund	Do-it-yourself print shop	●	●	●		
4.	5.	*The Newsroom: Clip Art Collection, Vol. 1*	Springboard	Additional graphics	●		●	●	●
5.		*Print Shop Companion*	Brøderbund		●	●	●		

Copyright 1986 by Billboard Publications, Inc. Compiled by the Billboard Research Department and reprinted by permission. Data as of 7/5/86 (entertainment) and 7/12/86 (education and home management).

NEW FOR YOUR ATARI®



Digitize your picture with the perfect hardware/software combination: COMPUTER EYES and MAGNIPRINT II.+ Digitize your voice with Parrot.

Computer Eyes lets you take any form of video input and saves it as a high-resolution graphics screen. You can use a video camera, VCR, TV output, video disk, other computers, etc. Now you can capture your picture, your friends or any video image and show it on an Atari computer Computer Eyes is an innovatve slow scan device that connects between any standard video source and your Atari computer (see the review in A.N.A.L.O.G magazine).

- Do a complete Hi-Res scan in under 6 seconds
- Unique multi-scan mode provides realistic grey scale images in 24 seconds, and up for more detail scans.
- Full one-year warranty on parts and labor
- Plugs into your Atari joystick ports and uses a standard video phono plug

Now anyone can create the kind of graphics seen in this ad When Computer Eyes is combined with Magniprint II+, you get unique capabilities that no other system can offer.

- Print your pictures in up to 19 different sizes, including poster sizes several feel across
- Print pictures with full shading for a level of realism even better than your TV screen
- Take your Computer Eyes images and modify them with your Koala Pad, Atari Touch Tablet, Micro Illustrator program, or Magniprint's special touch-up feature
- Works with Epson, Gemini, Panasonic, NEC, Citon, and other compatible dot-matrix printers (850 interface or equivalent required)
- Supports all Magniprint II + features

COMPUTER EYES, capture software and MAGNIPRINT II + ... Only $114.95

COMPUTER EYES/MAGNIPRINT Camera System
A complete ready to run system for those without access to video equipment. This system includes Computer Eyes, Magniprint II +, a high quality B/W video camera, and a 10 ft coaxial cable with appropriate connectors. Only **$299.95**

COMPUTER EYES alone (with capture and display software only) **$99.95**

Computer Eyes Gr.9 Acquisition Software..$12

MAGNIPRINT II+

"Magniprint II is a versatile and powerful program which will meet most, if not all your print out needs."
ANALOG MAGAZINE

Print your Atari® graphics like you've never seen before!

The plus is in the shading — now Magniprint lets you print the highest quality reproductions of your screens available anywhere It lets you choose the shading you want for the different colors and shades on the screen. It allows you to adjust the shades while your picture is displayed on the screen, so what you see is what you get Magniprint prints the shades on your standard printer and paper by using specially designed dot patterns, yielding better looking output than you get on your screen.

This new feature is perfect for printing pictures from Koala Pad, Touch Tablet, Micro Illustrator, and others It will amaze you when you print pictures from Computer Eyes or Strip Poker

Now get your pictures shaded the way you want them

Magniprint II is the most powerful print utility available for your computer! It is the only software that can accurately print GTIA modes with 16 shades pictures from PRINTSHOP* Prints 19 different sizes from ¼ page to giant wall sized posters Prints across or down the page, regular or inverse, with a centering option Works with printers compatible with NEC, CITOH, EPSON, & Gemini (850 interface or equivalent required) Accepts your own graphics creations or those created with Graphics Master*, Micropainter*, Fun With Art*, SYN Graph*, Atari Graphics Light Pen*, B/Graph*, Koala Pad* Atari Touch Tablet*, Paint*, Strip Poker*, Graphics Magician*, and others • Includes many beautiful sample pictures • Does amazing graphics with your standard printer and paper • Special feature lets you modify pictures with your joystick • Add text to your pictures • Can convert a graphics mode 9 picture to a graphics mode 8 one (and vice versa) • Allows you to print the whole screen or any selected portion of the screen in multiple sizes • Includes HELP screen for easy use • Prints GTIA modes 9 10, and 11 with 16 different shades Posters print along continuous sheets of paper which are attached together when done. 40% Faster ONLY $24.95

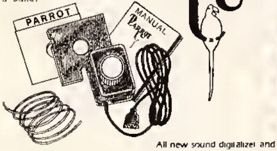
NOW prints text modes 0,1,2 with full Magniprint capabilities.

FREE. With every Magniprint II order get "PRINTALL" FREE PRINTALL prints your programs and files just as they appear on the screen. It clearly prints INVERSE and all the Atari graphics characters, and prints in regular or condensed print This alone is worth the price

DIGITIZE YOUR VOICE

PARROT...So good,
It is being used by
professional musicians
in a band!

PARROT



All new sound digitalizer and synthesizer for your Atari Tired of low-quality mechanical sounding voice output? Now you can make any Atari speak in your own voice Tired of four tone sound? Now any Atari can play a whole orchestra complete with a singing choir "The Parrot" digital sound synthesizer system lets you do all this and much more

How it works — "The Parrot" system plugs into your joystick port and lets you record pure digital sound from your stereo, TV, microphone or any other sound source The special Parrot software lets you play back this high quality sound on any Atari system with no special hardware needed It even lets you put this unbelievable sound right into your own programs, that will run on anyone's Atari It also includes digital sequencer software that lets you turn your Atari into a synthesizer comparable to those costing thousands of dollars Turn any natural sound into a musical instrument, or design your own custom sounds Imagine playing a song with the sounds of a dog's bark, a chinese gong, a car's honk, your own voice, or anything your imagination can come up with It turns your keyboard into an organ and lets you instantly switch between up to nine different digital sounds, each with three full octaves of notes Recording time varies depending on available memory and quality level desired You've got to hear it to believe it!!!!

THE PARROT digital input hardware and playback/synthesizer software with sample sounds and demos

All For Only **$39.95**

AT LAST A UTILITY THAT DOES IT ALL Scans & Analyses ALL Atari® programs Works on programs stored on Disk, Cartridge or directly from memory • Converts complex machine language into readable assembler • Transforms ANY Atari® BASIC program into

NEW AND IMPROVED FEATURES!

THE Scanalyzer

listable, modifyable BASIC • Changes a 4, 8 or 16K cartridge into a binary load file and source file that you can view & change using regular Atari® assembler. Clearly shows protection techniques such as. BAD SECTORS, BAD DATA MARKS, DUPLICATE SECTORS and FORCED CRC ERRORS. NEW FEATURES Now fully supports DOS 2 5 and includes one pass sector copier for 130XE owners Even finds and displays hidden directories No other program can do this!

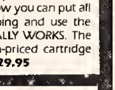
$29.95
Complete with instructions in theory and use.

IMPERSONATOR

CARTRIDGE TO DISK COPY SYSTEM

CARTRIDGE TO DISK COPY SYSTEM Yes, for only $29.95, you can make working copies of all your Atari computer cartridges (16K or less). Our special package will let you save your cartridges to ordinary disk files. They will run exactly like the originals when used with the Impersonator. Each disk holds up to 12 cartridge programs. Now you can put all your real cartridges away for safe keeping and use the Impersonator for everything. YES, IT REALLY WORKS. The Impersonator does everything the high-priced cartridge back-up systems do...and more. **ONLY $29.95**

SCHEMATIC DESIGNER



Schematic Oesigner is designed to help you create the electronic gadget of your dreams. Features 28 built-in symbols and the possibility of hundreds more, in high resolution graphics. You can add titles, zoom in and print out your schematics. SCHEMATIC DESIGNER includes many sample schematics which can be used to build many projects, including a working light pen.

SCHEMATIC OESIGNER ON A OISK... Only $24.95

ATARI SOFTWARE PROTECTION TECHNIQUES
ADVANCED ATARI PROTECTION TECHNIQUES
GEORGE MORRISON

BOOK I + DISK: (The Original) Thoroughly explains the techniques used by advanced software pirates, and the copy protection methods used to stop them. It offers clear and understandable explanations sophisticated enough for software writers of any scale yet easy enough for a beginner just wanting to learn more about Atari® computers. A MUST READ FOR ALL ATARI® OWNERS

BOOK INCLUDES: • Duplicate sectoring • Custom disk formatting • Creating "BAD" sectors • Hardware data keys • Legal protection like copyrights, trade secrets, patents • Protecting BASIC programs • Self-modifying Code • ROM + EPSOM cartridges • Hidden serial numbers • Self-destructing programs • Freeware • Misassigned sectoring • Much, much more

DISK INCLUOES: • Directory mover • VTOC scanner • Duplicate sector finder • Sector mover • Bad sector writer • Sector data displayer • Autorun builder • Other useful programs.

This comprehensive book and disk package should not be confused with low quality imitations offered elsewhere

BOOK II + OISK II: Advanced Software Protection This all new sequel starts where the highly acclaimed Book I leaves off Book II is the most up-to-date resource available for the Atari® owner. Includes reviews and explanations of products such as. The Happy Enhancement,* The Impossible,* The Scanalyzer,* The Chip,* The Pill* and Super Pill* & many others.

Book fI: Tells you specifically what they copy, what they won't, how they are used, and the details of how they work Book II also includes such topics as. • Transmitting protected programs • Copying disks with more than 19 sectors/track Includes the newest protection methods by companies like Synapse* AND Electronic Arts* • Data encryption • Phreaking methods • Program worms • Logic bombs • Bank-select cartridges • Random access codes • New trends in software law • Sample BASIC + Assembler programs • On-line security • And much more.

DISK II INCLUDES: • Automatic program protector • Custom format detector • Newest protection demos • Forced password appender • Data encrypter • And much more

Book + Disk Packages only $24.95 each or Special Offer both for only $39.95

ALPHA SYSTEMS

MAIL TO ALPHA SYSTEMS – Dept. C
4435 Maplepark Rd./Stow, OH 44224 Send check or money order Include $2.00 shp & hdlg Ohio residents 5½% sales tax
CALL 216-374-7469 to charge to MasterCard or VISA

BONUS: Order any 3 programs and get FREE your choice
□ Deluxe Space Games (3 games on a disk) or □ Disk Pak 1000 (Utility Package)

All for your Atari Computers. Disk drive and 48K required. Atari® is a registered trademark of Atari Corporation.
*Denotes products not related to Alpha Systems

ALL PRODUCTS WORK WITH ALL ATARI 8 BIT COMPUTER SYSTEMS

Promoting Computers In School

Kathy Yakal, Assistant Features Editor

Via free or discounted hardware and software, along with special teacher training, computer hardware manufacturers continue to promote their microcomputers in schools at every level. Here's an overview of recent efforts to increase the already impressive penetration of this technology into classrooms across the land.

Microcomputers now play a significant role in many areas of education. But getting computers into the classroom and deciding how they are best used continue to be subjects of much debate. A combination of factors has slowed the process even further: the problems of implementing a new, evolving technology; the chaotic atmosphere of the computer industry itself; the computer education of teachers and administrators; and the relatively tight budgets of educational institutions.

Nevertheless, tremendous changes have occurred in teachers' attitudes toward microcomputers over the last couple of years. There are several reasons. First, software publishers have increasingly attempted to provide the kind of programs that teachers feel comfortable with—quantifiable, curriculum-based software. At the same time, innovative, nontraditional kinds of learning aids have gained a wider acceptance. Second, the hardware and software shakeouts that have moved the computer industry toward maturity and greater stability have made educators feel more confident about making a financial commitment to microcomputers. Finally, teachers are generally less anxious about computers and more experienced at applying them, with a growing number of classroom success stories fueling increased computer use. It's not just the students and a few computer-wise teachers who are driving the movement anymore.

Each of the major computer manufacturers has made unique contributions to trigger the integration of computers into classrooms. Some offer educational discounts. Others provide special grants and develop efficient ways to exploit the hardware, such as networking. In addition to easing the financial burden, hardware manufacturers promote the general health of the educational computing industry by fostering quality software development and encouraging nontraditional applications of hardware to traditional curricula. Inservice training of teachers and special workshops sponsored by hardware companies have also been significant in creating a more upbeat attitude toward classroom computing in recent years.

Here's a company-by-company look at the variety of approaches.

Apple Computer

Officials at Apple Computer realized early on that a good software base was central to getting their hardware into schools. Apple made major efforts in the early 1980s to convince software developers to support its machines, offering them shared advertising, discounts on development machines, and technical support.

Currently, Apple has two educational discount programs. *Step pricing* gives buyers lower prices on larger orders, encouraging educators to buy in quantity whenever possible. And with the *Volume Purchase Agreement*, a school can elect to pay for its computers over a three-year time period. If a school involved in such an agreement finds that the hardware does not meet its needs, it may return the equipment without making the remaining payments.

Support after the sale is also a key to Apple's success in the school market. Apple relies heavily on its local dealers to provide on-site support to educators. Ten days before an order of computers is scheduled to reach a school, Apple notifies a local dealer who is then responsible for installing the equipment and providing orientation and training for teachers and administrators. The dealer is also responsible for any follow-up repair and maintenance.

Apple has developed a fairly high profile on many college campuses across the country, thanks to the Apple University Consortium (AUC). A couple of years ago, 24 U.S. colleges and universities formed an organization whose purpose was to develop tools and resources for the Macintosh. Because of that, many campuses today maintain busy Macintosh labs and workstations. At least one institution, Drexel University, *requires* its freshmen to purchase Macintoshes.

Atari Corporation

Atari Corporation's change of ownership and revamped management have resulted in few formal educational programs currently in operation. Considering Atari's growing strength, however, that may soon change. Low-cost 8-bit Ataris have already been the first kind of computer many students ever encountered in a class; their current availability and strong software base may even amplify this trend. And the low price of the powerful ST computers, as well as their strong graphics and music capabilities, may cause some educators to look twice, especially for use in creative applications.

Atari recently announced a marketing agreement with Montreal-based Arrakis, publisher of the *Advantage* series of educational software. ST versions of these programs, which have in the past been available for Apple, Commodore, and IBM, should be ready by the end of the year. The Arrakis series is known for its impressive graphics and cartoonlike animation, as well as a sophisticated parser which incorporates principles of artificial intelligence and provides direct answers to students' questions.

Computer Curriculum Corporation (Palo Alto, CA) has announced a commitment to Atari equipment. CCC is packaging STs along with their minicomputers and a series of courses; that is, they bundle hardware and software and install the complete systems in schools.

Finally, a 10-percent discount is available to colleges and universities, with follow-up service and support provided by local dealers.

Commodore

Commodore's big draw for schools lies in its inexpensive hardware and broad base of third-party educational software. Many teachers, unable to get funds allocated for major hardware purchases, started out by buying a few Commodore 64s (or even bringing their own in from home). In many settings, this was all that was necessary to get students familiar with the fundamentals of microcomputers, while also providing workstations for word-processing, database management, and computer-aided learning. In other cases, some school administrators have been willing to make a financial commitment to microcomputers in the classroom, based on the excitement they've seen generated by a few hundred dollars' worth of hardware and software.

Every major educational software publisher supports Commodore machines, so hundreds of titles have been developed for the Commodore 64 over the last few years. Though some are more appropriate for the less structured atmosphere of the home, many have been adopted for classroom use. A complete list of the more than 1500 packages will be available through distributors this fall.

Commodore has recognized that computer-aided education does not necessarily have to happen in a schoolroom, and has supported some unique opportunities for learning. Two of these involve telecommunications. Quantum-Link, a year-old service that Commodore has backed with technical and marketing assistance, is an online forum for sharing information of all kinds. Though much of the earliest activity that went on there was computer-oriented, a variety of other special interests are now supported there. Education is one of them. The Resource Center, a relatively new forum in the Learning Center area of Q-Link, is composed of three sections. The Library includes curriculum guides, teaching strategies, software reviews, and articles about home and community education. In the Media Room, users can download software written

Each of the major computer manufacturers has made unique contributions to trigger the integration of computers into classrooms.

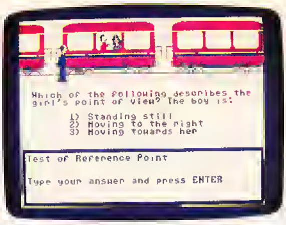
by teachers. The Lounge is an online conference area, a meeting place for teachers and parents to gather and discuss educational issues and plans. And the Resource Center's Message Boards keep everyone posted on what's happening in educational computing. (Quantum Computer Services, 8620 Westwood Center Dr., Vienna, VA 22180.)

Commodore is involved with another online educational venture: the Electronic University Network, operated by TeleLearning Systems, Inc., of San Francisco. By purchasing the $195 enrollment package, you have access to online courses offered by 25 colleges and universities. You may either take selected courses or, if you have met the school's prerequisites, work toward an M.B.A. or undergraduate degree. Degrees are issued by the schools involved, not by the Electronic University Network. The system software also gives you access to online databases—libraries of information for research purposes—as well as counseling and online seminars. (Software allowing IBM and Apple owners to use the network is also available. For more information, write to TeleLearning Systems, Inc., 505 Beach St., San Francisco, CA 94133, or call (800)22LEARN; in California, call (800)44LEARN.)

Commodore has, in the past, participated in more traditional outreach efforts to schools. Recent financial problems at the company have apparently forced cutbacks in ongoing educational support. That, too, may change if Commodore is able to weather remaining financial hurdles. The company has a strong history of major support to Canadian schools, and continues to maintain that presence.

IBM

IBM has made a major commitment to the basic skills of reading and writing with its Writing To Read program in the school market. Developed by educator Dr. John Henry Martin, Writing To Read was tested among 22,000 students and was evaluated in an independent two-year study by the Educational Testing Service before being introduced in the fall of 1984. The program has grown in use from 200 schools at the end of 1984 to 1100 schools at the end of 1985. More than 125,000 students have participated in the program. The computer-based program allows students to advance at their own pace and offers positive reinforcement during a student's interaction with the computer.



```
Which of the following describes the
girl's point of view? The boy is:

        1) Standing still
        2) Moving to the right
        3) Moving towards her

Test of Reference Point

Type your answer and press ENTER
```

Atari recently announced that 17 titles from the acclaimed Arrakis series will be available for the ST.

Through Writing To Read, children learn the 42 *phonemes* (letter and sound combinations) that make up the English language. Using these phonemes, students are able to read and write everything they can say. Typically, students spend an assigned hour each day in a Writing To Read center or lab, a specially designed room made up of five learning stations. Work sessions in the lab are generally an hour long. Students alternate around the five stations: at the computer, with a work journal, at a listening library using specially taped lessons, and playing two phoneme-based games at the "make word" station.

The Tandy 1000 computer is becoming an increasingly popular choice for educators.

IBM has made a significant commitment to developing curriculum-based software in many subject areas for elementary and secondary schools, programs that come bundled with several student disks and a teacher's guide for easy use in classrooms with multiple computer workstations. Many of the programs are also available individually. In addition, IBM has founded the National Disability Resource Center, a national technology resource that supports the needs of the disabled.

Tandy Corporation/ Radio Shack

The Tandy Corporation has had a longstanding commitment to computer use in the schools. In 1979, Tandy introduced the first low-cost classroom network system—Network 1. In 1980, the Radio Shack Education Division was formed to produce a line of educational courseware. In the years since, Tandy has offered free computer literacy training to teachers, provided formal support for educational software publishers, donated more than $1 million in hardware and software products to support research and development activities, and sponsored conferences and associations to promote the further integration of computers into classrooms.

Currently, three major programs are in place in addition to these areas of ongoing support. In conjunction with Education Systems Technology Corporation (ESTC), Tandy offers an integrated learning system for elementary schools, consisting of three major components: a comprehensive 1500-lesson reading and mathematics curriculum for grades K-6; a computer laboratory composed of 1 Tandy 3000 host computer and up to 40 Tandy 1000 personal computer workstations, allowing an entire class to use the system at once; and an on-site facility management service, which includes an ESTC lab attendant and a complete computer-controlled student management and performance reporting system.

*"BEST ADVENTURE GAME 1985" COMMODORE COMPUTING INTERNATIONAL

CHARTBUSTERS

Frankie have sent you over 60 tasks in your journey from Mundanesville through the Pleasure Dome. Tasks ranging from the trivial, to heroic feats of skill and intelligence. Whenever you complete these challenges a bar-chart will show your increase in the various elements of your personality and Pleasure points will be awarded... This combination awards you the minimum requirement (99% a *complete person*) to search for the Special Door – the Door to the Ultimate Experience – the heart of the Pleasure Dome.
Your goal is to become a *complete person* and to achieve this you must reach the top on the Bar Charts (when the word BANG will light up above the personality factors) and achieve a score of 87,000 Pleasure Points.

"Best adventure game 1985" Commodore Computing International

FIREBIRD LICENSEES INC. P.O. BOX 49, RAMSEY, N.J. 07446   FACTSHEET/ORDER HOTLINE: 201.934.7373

COMMODORE 64/128 ARE TRADE MARKS OF COMMODORE BUSINESS MACHINES.

GOLD DISK
COMMODORE
64/128K

Lyco Computer Marketing & Consultants

ATARI

ATARI

1050	129
SF314	219
SF354	175
130XE	CALL
65XE	CALL
520st	CALL
520st monochrome	CALL
520st color	CALL
1027 printer	145
1040st (NEW)	CALL

BRODERBUND (Atari)

Printshop	28.75
Graphics Lib I, II, III	18.75
Paper refill	12.95
Karateka	19.75
Printshop comp	24.75

SSI (Atari)

NAM	24.75
Mechbrigade	34.95
Antietam	29.95
U.S.A.A.F.	34.95
Col. Conquest	24.75
Ambush	34.95

SUBLOGIC (Atari)

Flight Simulator	29.95
Night Mission Pinball	18.95
Scenery Disks EA	14 95

SYNAPSE (Atari)

Synfile	29.95
Syncalc	29.95
Template	14.95

ACCESS (Atari)

Leader board	24.75

FIREBIRD (Atari)

The pawn	26.75
Star glider	26.75

ACTIVISION (Atari)

Hacker	15.75
Mindshadow	15.75
Ghostbusters	15.75
Great Am Race	15 75
Music Studio	22 75
Space Shuttle	15 75

UNISON WORLD (Atari)

Printmaster	24.75
Art Gallery	18.75

MICROLEAGUE (Atari)

Baseball	24.95
GM disk	24 95
Team disk	14.95

ACTIVISION (520 st)

Borrowed Time	29.75
Music Studio	34.75
Hacker	26.75
Little People	29.75

VIP (520 st)

VIP Professional	109
VIP Lite	65.95

HABA (520 st)

Writer	35.95

apple

ACTIVISION (Apple)

Alter Ego	28.75
Little People	24.75
Mindshadow	24.75
Hackler	24.75
Gamemaker	24.75

BRODERBUND (Apple)

The Print Shop	31.50
Graphic Librarys EA	18.50
Bank St. Writer 128K	42.75
Bank St. Speller	42.75
Carmen Sandiego	22.75
Karateka	22.75
Captain Goodnight	22.75
Muppet Cruise	25.75
P.S. Companion	24.75
Science Kit	35.95

MICROPROSE (Apple)

Crusade in Europe	24.75
Decision in Desert	24.75
F-15 Strike Eagle	20.75
NATO Commander	20.75
Silent Service	20.75
Solo Flight	20.75

SSI (Apple)

Phantasie II	24.75
Wizard's Crown	24.75
Rings of Zilfin	24.75
Colonial Conquest	24.75
Battlegroup	35.75
NAM	29.75

MICROLEAGUE (Apple)

M L Baseball	24.95
General Mgr	24.95

COMMODORE

COMMODORE

128	CALL
C 1571 Drive	CALL
C 1902-A	CALL
C 1541 Drive	CALL
C 1670 Modem	CALL
C 1350 Mouse	39
C 1700 128 K RAM	145
C 1750 512 K RAM	269
JANE	35
Perfect Writer	45
Perfect Calc	45
Perfect Filer	45

BATTERIES INCLUDED

Paper clip	35.95
Consultant	35.95
Paper clip w/ spellpak	48.95

EPYX-64

Movie Monster	24.75
Vorpol Kit	22.75
Fast Load	24.75
Summer Games II	24.75
Super Cycle	24.75
World Games	24.75
Football	24.75

SUBLOGIC

Nightmission Pinball	18.95
Flight Simulator	29.95
Jet Simulator	25.95
Football	25.95
Scenery Disk EA	14.95
Set 1-6	69.95

CARDCO

CB/2 2-slot Board	19.95
S'More Basic Rom	39.95
Freeze Frame	28.75

ACTIVISION (C-64/128)

Alter Ego	28.75
Hacker	18.75
Little People	20.75
Game Maker	24.75
Borrowed Time	18.75
Space Shuttle	18.75
Music Studio	22.75
Mind Shadow	18.75
Road Race	18.75
Fast Tracks	22.75
Count Down	18.75
Basketball	18.75

BRODERBUND

The Print Shop	24.75
Graphics Library I, II, III	15.75
Karateka	17 75
Bank St. Writer	29.75
Lode Runner	19.75
Printshop Companion	24.75
Bank St. Speller	29.75
Bank St. Filer	29.75
Bank St. Mailer	29.75
Championship Loderunner	19.75

UNISON WORLD

Print Master (Amiga)	22.75
Print Master (C-64/128)	22.75
Art gallery	16.75

FIREBIRD

Elite	19.95
Colossus IV	21.95

ACCESS (C-64-Amiga)

Leader Board	24.75
Leader Board Amiga	24.75
Tournament I	12.75
MACH 128	28.75
10th Frame	24.75

XETEC

Font Master II 64	28.95

MODEMS

DIGITAL DEVICES

Pocket Modem AT	Call
Compuserve	*18.95*

ZOOM

PC 1200 ST	159
PC 1200 XL	259
PC 1200 XL/PFS	279

US ROBOTICS

Password 1200M	189
Courier 2400	395

SUPRA

1200 (520 ST)	149

DSI

Messenger 64/128	39.95

SUPRA

1064 Modem (C-64)	49.95

ANCHOR

Volksmodem	55
Volksmodem 12	179

COMPUSERVE . . 18.95

COMMODORE

1670 Modem	155

MONITORS

PANASONIC

TR-122 MYP 12" Amber TTL	139
TR-122 M9P 12" Green TTL	139
TX-12H3P 12" RGB	369
DT-H103 10" RGB	349

TEKNIKA

MJ-10	149
MJ-22	249
MS-305 RGB	309

THOMPSON

365 12 RGB	CALL

HITACHI

MM-1218 12" Green	99
MM-1220 12" TTL Amber	129
CM-1406C 13" color w/ cable	179
CM-1409 13" RG8	305
CM-1216D 12" RGB	385
CM-1455S 13" 720x350	525
CM-1457A 13" RG8 720x460	679

ZENITH

ZVM 1220	89
ZVM 1230	89
ZVM 1240	149

NEC

Multisync	CALL

PRINCETON GRAPHICS

MAX-12 Amber	175
HX-12 RGB	458
SR-12 RGB	575

NEW HOURS!
Mon-Thur 9AM-8PM
Fri 9AM-6PM
Sat 10AM-6PM

LYCO COMPUTER MARKETING & CONSULTANTS, INC.

NEW HOURS!
Mon-Thur 9AM-8PM
Fri 9AM-6PM
Sat 10AM-6PM

Lyco Computer Marketing & Consultants

AMERICA'S MAIL ORDER HEADQUARTERS

NEW HOURS!
Mon-Thur 9AM-6PM
Fri 9AM-6PM
Sat 10AM-6PM

NEW HOURS!
Mon-Thur 9AM-8PM
Fri 9AM-6PM
Sat 10AM-6PM

SAVE ON THESE IN STOCK PRINTERS

1080 ..... $199

PANASONIC

1080	195
1091	225
1092	309
3131	249
3151	399
1592	419
1595	599

EPSON

LX80	209
FX85	CALL
DX10	207
HI80	355
HS80	298
FX286	CALL
LO800	529
LO1000	659

SEIKOSHA

SP-1000 VC (C-64)	175
SP-1000 A Centronics	195
SP-1000 I IBM	195
SP-1000 AS RS-232	195
SP-1000 AP Apple IIc	195
BP-5200 I	649
BP-1300	469
BP-5420	999
SP-1000 ribbon	8.50
BP-5420 ribbon	12.50

SILVER REED

EXP 420 P	209
EXP 600 P	489
EXP 800 P	649
EXP 400	249
EXP 770	749

LEGEND

1080	Call
1380	258
1385	289
808	148

OKIDATA

Okimate 10	179
182	214
192	348
193	563
Okimate 20	199
120	205
292	CALL
293	CALL

C. ITOH

1550 SP+	Call
D1040	Call
Prowriter Jr.	Call
Prowriter 8510 SP+	Call

TOSHIBA

P351+	1149
P341P	969
P341S	999
351 sheet feeder	529
321 P/S	495

JUKI

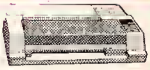
Juki 6100	339
RS 232 Serial Board	55
5510	349
Color Kit	105
6100 Tractor	119
6100 Sheet Feeder	209
Juki 6300	757

CITIZEN

Premier 35	469
MSP-10	285
MSP-15	385
MSP-20	325
MSP-25	485
120-D	179

DIABLO

D25	549
D801F	2395
P 32 CO1	699
P-38	1749
635	1029



NX-10 .... CALL

STAR MICRONICS

NX-10 (NEW)	CALL
NB-15 (NEW)	CALL
SB-15 (NEW)	CALL
NL-10 (NEW)	CALL
SG-15	367
SD-10	319
SD-15	438
SR-10	469
SR-15	578
SB-10	589
Powertype	297

BROTHER

HR-15XL-P	359
HR-15XL-S	359

COLOR RIBBONS NOW AVAILABLE!!

DUST COVERS

Atari

520ST	11.95
130XE	6 99
800XL	6 99
1050	6.99
1025	7.99

Commodore

C128	7 99
1571/1541	6 99
1902	10 95
1702	8 99
C64/Vic20	6.99

Panasonic

1090/1091	8 99
1092	8 99
1093	9 99

DRIVES

INDUS

GT Atari	179

INDUS

GT Commodore	179

ATARI

1050	129

COMTEL

Enhancer 2000 (C-64)	149

LOWEST PRICES

INTERFACING

CARDCO

G-WIZ (C-64)	47
Super G (C-64)	54
C/?PS (C-64)	37

ORANGE MICRO

Grappler + (Apple)	85
Grappler 16K (Apple)	149
Orange (Apple)	59
Grappler C (IIc)	85

MICROBITS

MPP-1150 (Atari)	45
MPP-1150 XL (Atari)	49
Microprint (Atari)	35

ATARI

850	109

DIGITAL DEVICES

U-Print A 16	45
Apeface	39

XETEC

Super Graphix 64	64
Super Graphix JR 64	45

SUPER SAVINGS

DISKETTES

5¼'' DISKETTES

MAXELL

SSDD	4.99
DSDD	12.99

VERBATIM

SSDD	9.50
DSDD	12.99

BONUS

SSDD	6.99
DSDD	7.50

SUNKYOUNG

SKC SSDD	11.99
SKC DSDD	13.99

3.5'' DISKETTES

3M

SSDD	16.99
DSDD	25 95

MAXELL

SSDD	16.99
DSDD	23.99

VERBATIM

SSDD	16.99
DSDD	24.99

DISK NOTCHERS . . $7.95!!

FASTEST SERVICE

IBM-PC

BRODERBUND (IBM)

Bank St. Writer	48.95
Print Shop	34.95
Graphics Lib. 1	22.95
Graphics Lib. 2	22.95
Ancient Art of War	22 95
Champ Lode Runner	22.95
Karateka	22.95

ACTIVISION (IBM)

Borrowed Time	24.75
Mindshadows	24.75
Music Studio	29.95
Alter Ego	29.95
Alter Ego (Male)	29 95
Pebble Beach	29.95

SUBLOGIC (IBM)

Jet Simulator	34.95
Scenery Disks EA	14.95
Set 1-6	69.95
S.D. Japan	14.95
S.D. San Francisco	14.95

SYNAPSE (IBM)

Synslock	64 95
Essex	28.95
Wizard of Wall St	28 95
Brimstone	28 95

QUADRAM

Gold quadboard	399
Silver quadboard	229
Ouadboard	199

TOLL FREE 1-800-233-8760

In PA 717-494-1030
Customer Service 717-494-1670

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA
17740

MasterCard VISA

RISK FREE POLICY

In stock items shipped within 24 hours of order No deposit on C O D orders Free shipping on prepaid cash orders within the continental U S Volume discounts available PA residents add sales tax APO FPO and international orders add $5.00 plus 3% for priority mail service Advertised prices show 4% discount for cash, add 4% for MasterCard and Visa Personal checks require 4 weeks clearance before shipping We do not guarantee compatibility We only ship factory fresh merchandise Ask about UPS Blue and Red label shipping All merchandise carried under manufacturer's warranty Return restriction applicable Return authorization required All items subject to change without notice

Tandy offers special pricing on educational network systems. Educational customers purchasing two Model 4 computer/Network 4 Student Stations at the regular price of $1,099 each will receive a third station free. Other network configurations are available at substantial discount. Network 4 is Tandy's newest and most powerful classroom network. Students at individual stations can sign on to teacher-created network accounts to share software stored on the hard disk drives at a central disk server. The system supports multiple disk and printer servers, which saves money by eliminating the need for peripherals at each workstation. (This promotion is good through September 30, 1986; interested customers may contact a Regional Education Coordinator by calling 800/433-5628.)

Finally, topics for the third and fourth quarter Grants Program have been announced. All nonprofit educational institutions and professional educators are eligible to submit proposals for these project grants. Proposals for "Creative Uses of Microcomputers in Education" should be submitted by September 30, 1986, and proposals for "Using Computers for Instructional Management" should be submitted by December 31, 1986. (Information packets required for use in order to submit proposals can be obtained by writing to Tandy Educational Grants Program, 1400 One Tandy Center, Fort Worth, TX 76102.)

For further information on any of the products or programs mentioned here, please contact:

Apple Computer
20525 Mariani Ave.
Cupertino, CA 95014

Atari Corporation
1196 Borregas Ave.
P.O. Box 3427
Sunnyvale, CA 94088

Commodore Business Machines
1200 Wilson Dr.
West Chester, PA 19380

IBM Educational Systems
P.O. Box 2150
Atlanta, GA 30055

Tandy Corporation/Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102

THE REFERENCE *Library* OF THE FUTURE

Kathy Yakal, Assistant Features Editor

Traditional classroom education has already undergone some major changes with the continuing integration of microcomputers into schools. But there's a relatively new technological development with far-reaching educational implications—CD-ROM (Compact Disc-Read Only Memory). By connecting a personal computer to a compact disc containing digital information, you can easily store and cross-reference an entire encyclopedia, with plenty of room to spare. Similar to the laser-driven audio compact discs that now hold an hour or so of recorded music, these new computer peripherals will surely alter many of our current approaches to education. Here's a look at what this might mean for the classroom of the future.

Your grandchild's sixth-grade history homework assignment: Turn in a report on the first manned space flight to the moon. Though the topic may sound typical, the research won't involve trudging to the school library or home encyclopedia to haul down 15 different books and stare at reams of text and a static photo of the moon.

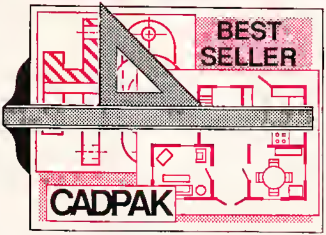
Instead, with a few keystrokes:

- A laser-driven compact disc feeds through a computer system the digitized voice of Neil Armstrong as he steps down onto the moon's surface—*That's one small step for ....*
- A full-color animated sequence shows the touchdown of the lunar lander and the planting of the American flag. Another sequence displays the liftoff, orbital rotation, and space flight while the process is explained through speech synthesis and onscreen text.
- A few more keystrokes and the voice of President John F. Kennedy can be heard as, earlier in the decade, he set a national goal to reach the moon.
- A series of articles—all with voice, color animation, and printout

SUPER SOFTWARE

BASIC Compiler
BASIC
Give your BASIC programs the speed and performance they deserve

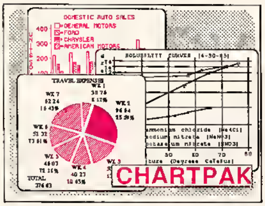
The *complete* compiler and development package. Speed up your programs 5x to 35x. Many options: flexible memory management; choice of compiling to machine code, compact p-code or both. '128 version: 40 or 80 column monitor output and FAST-mode operation. '128 Compiler's extensive 80-page programmer's guide covers compiler directives and options, two levels of optimization, memory usage, I/O handling, 80 column hi-res graphics, faster, higher precision math functions, speed and space saving tips, more. A great package that no software library should be without. **128 Compiler $59.95**
64 Compiler $39.95



Remarkably easy-to-use interactive drawing package for accurate graphic designs. New dimensioning features to create exact scaled output to all major dot-matrix printers. Enhanced version allows you to input via keyboard or high quality lightpen. Two graphic screens for COPYing from one to the other. DRAW, LINE, BOX, CIRCLE, ARC, ELLIPSE available. FILL objects with preselected PATTERNS; add TEXT; SAVE and RECALL designs to/from disk. Define your own library of symbols/objects with the easy-to-use OBJECT MANAGEMENT SYSTEM–store up to 104 separate objects. **C-128 $59.95**
C-64 $39.95

For school or software development. Learn C on your Commodore with our in-depth tutorial. Compile C programs into *fast* machine language. C-128 version has added features: Unix™-like operating system; 60K RAM disk for fast editing and compiling Linker combines up to 10 modules; Combine M/L and C using CALL; 51K available for object code; Fast loading (8 sec. 1571, 18 sec. 1541); Two standard I/O librarys plus two additional libraries—math functions (sin, cos, sqrt, etc.) & 20+ graphic commands (line, fill, dot, etc.).
C-128 $59.95
C-64 $59.95

Not just a compiler, but a complete system for developing applications in Pascal with graphics and sound features. Extensive editor with search, replace, auto, renumber, etc. Standard J & W compiler that generates fast machine code. If you want to learn Pascal or to develop software using the best tools available–**SUPER Pascal** is your first choice.
C-64 $59.95



Easily create professional high quality charts and graphs without programming. You can immediately change the scaling, labeling, axis, bar filling, etc. to suit your needs. Accepts data from CalcResult and MultiPlan. C-128 version has 3X the resolution of the '64 version. Outputs to most printers.
C-128 $39.95
C-64 $39.95

OTHER TITLES AVAILABLE:

COBOL Compiler

Now you can learn COBOL, the most widely used commercial programing language, and learn COBOL on your 64. COBOL is easy to learn because its easy to read. COBOL Compiler package comes complete with Editor, Compiler, Interpreter and Symbolic Debugger. **C-64 $39.95**

BEST SELLER

Personal Portfolio Meneger

Complete portfolio management system for the individual or professional investor. Easily manage your portfolios, obtain up-to-the-minute quotes and news, and perform selected analysis. Enter quotes manually or automatically through Warner Computer Systems. **C-64 $39.95**

Xper

XPER is the first "*expert system*" for the C-128 and C-64. While ordinary data base systems are good for reproducing facts, XPER can derive knowledge from a mountain of facts and help you make expert decisions. Large capacity. Complete with editing and reporting. **C-64 $59.95**

PowerPlen

One of the most powerful spreadsheets with integrated graphics. Includes menu or keyword selections, online help screens, field protection, windowing, trig functions and more. PowerGraph, the graphics package, is included to create integrated graphs and charts. **C-64 $39.95**

Technical Analysis System for the C-64	$59.95
Ada Compiler for the C-64	$39.95
VideoBasic Language for the C-64	$39.95

C-128 and C-64 are trademarks of Commodore Business Machines Inc.
Unix is a trademark of Bell Laboratories

Abacus Software

P.O. Box 7219 Dept. C9 Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Call **now** for the name of your nearest dealer. Or to order directly by credit card, MC, AMEX of VISA call **(616) 241-5510**. Other software and books are available–Call and ask for your **free** catalog. Add $4.00 for shipping per order. Foreign orders add $12.00 per item. Dealer inquires welcome–1400+ nationwide.

capabilities—reveal the early attempts at space flight, including a revolving three-dimensional overview of Sputnik; the voice of rocket expert Werner Von Braun; a cross-section of a typical rocket system, revealing how the physical configurations have changed over time; and a brief explanation of early V-1 and V-2 rockets during World War II.

Dozens of additional topics offer themselves almost magically to the young researcher—from Andy Williams singing *Moon River* to an animated demonstration of the moon's effects on the Earth's tides.

Although such examples may sound farfetched today, the development of this technology is already under way. The *interactive* nature of research in tomorrow's schools will be a far cry from the traditional approach.

For schoolchildren today, finding information is, in many ways, similar to the process that was followed by their parents and grandparents. The millions of available books can be a fascinating but often frightening and frustrating world for young students. And cross-referencing information from one source to another is even more daunting. The search process itself can sometimes be discouraging enough to thwart many students' early efforts at learning.

In the next few years, however, laser technology in the form of compact disc players interfaced with personal computers are expected to have a major impact on how students research. Called CD-ROM, this configuration of digital technology embodies three elements that offer tremendous power for educational research. First, speed: Using a CD-ROM system, a student can find the most trivial fact contained in a multivolume reference work in the time it would take to remove a book from the shelf and flip it open to the index. Second, durability: Because the search functions of CD-ROM are driven by a laser beam reading a disc, the hardware and software, given reasonable care, could last hundreds of years. And third, tremendous storage capability: A compact disc can hold over 550 *megabytes* of data. That's roughly a quarter of a million pages of text on a disc smaller than a 45 rpm record.

A Long Time Coming

The power of lasers was harnessed over twenty years ago and has potential applications in many industries. Engineers at many consumer electronics companies worldwide have been experimenting with consumer and business applications for almost as long as the technology has been available. We saw some of the first results of this experimentation in 1980, when Sony and N.V. Philips of the Netherlands announced specifications for a new kind of home stereo system: *compact disc–audio.* Compact disc players use laser beams to read music digitally encoded in microscopic pits on the disc. Since nothing actually touches the disc itself in the playing process, there is no wear on the disc. And the recording is free of the hisses and pops and other distortions we've grown accustomed to hearing on albums. CD players began appearing on the market in 1983 and, thanks to market acceptance, are now a very reasonably priced alternative to traditional stereo systems.

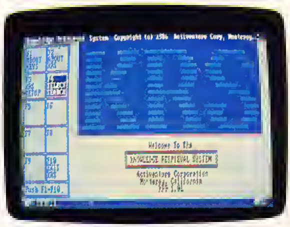
In that same year, Sony and Philips announced specifications for another way to use CD technology: Compact Disc–Read Only Memory (CD-ROM). Slightly modified CD players interfaced with personal computers are capable of holding the data that would require hundreds of the floppy disks that we've grown accustomed to using for data storage. And with the right search software, access to that data is almost instantaneous.

Reference material is an obvious first application for CD-ROM. Consequently, the first hardware/software configuration actually available for the consumer market was a joint venture between Philips, which provided the player, and Grolier Electronic Publishing, which offered its online *Academic American Encyclopedia* on a compact disc. The package, sold in limited outlets across the country, retails for $1,495.

Amazing Searches

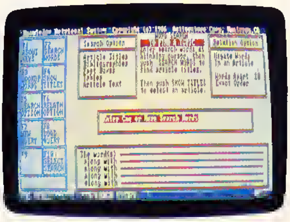
Many now claim that the CD-ROM is superior to any previous reference tool. To see why, let's take a brief walk through a search using the Philips/Grolier package.

Installation of the system involves plugging a board into the IBM-PC, connecting the CD player cable to the PC, and turning everything on. Once you've loaded the search software (*Knowledge Retrieval System*, by Knowledge Set) from a floppy disk, put the CD into the drive and turned it on, you're ready to go.



Here is the opening screen of the CD-ROM search software developed by Knowledge Set (formerly Activenture).

The opening screen offers you the options of finding out more about the system itself, moving directly into a search, or entering the system. All commands are issued by simply pressing the desired function key.



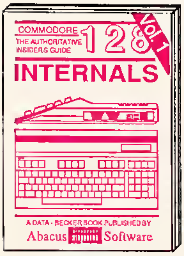
Enter search words at blinking cursor, then push SEARCH WORDS to find article titles.

Then push SHOW TITLES to select an article.

Step 1: Set your search and relation parameters and enter the words or phrases you want to explore.

The first working screen of the system presents two sets of options. Search options let you look for desired words or phrases within article titles, bibliographies, fact boxes, article text itself—or anywhere in

Top shelf books from Abacus


COMMODORE 128 Vol.1
THE AUTHORITATIVE INSIDERS GUIDE
INTERNALS
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

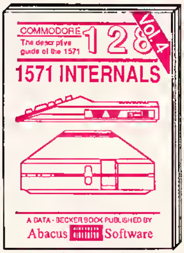
Detailed guide presents the 128's operating system, explains graphic chips, Memory Management Unit, 80 column graphics and commented ROM listings. 500pp $19.95

COMMODORE 128 Vol.2
Complete BASIC 7.0 ROM listings
BASIC 7.0 INTERNALS
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

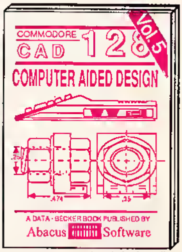
Get all the inside information on BASIC 7.0. This exhaustive handbook is complete with commented BASIC 7.0 ROM listings. Coming Summer'86. $19.95

COMMODORE 128 Vol.3
Practical & easy to use techniques for C-128
TRICKS & TIPS
BEST SELLER
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

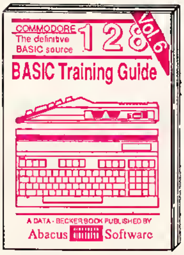
Filled with info for everyone. Covers 80 column hi-res graphics, windowing, memory layout, Kernal routines, sprites, software protection, autostarting. 300pp $19.95


COMMODORE 128 Vol.4
The descriptive guide of the 1571
1571 INTERNALS
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

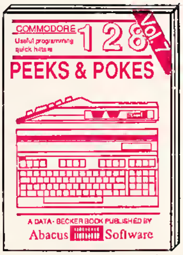
Insiders' guide for novice & advanced users. Covers sequential & relative files, & direct access commands. Describes DOS routines. Commented listings. $19.95



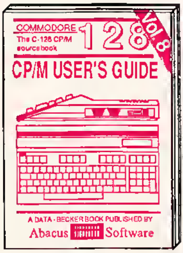
Learn fundamentals of CAD while developing your own system. Design objects on your screen to dump to a printer. Includes listings for '64 with Simon's Basic. 300pp $19.95


COMMODORE 128 Vol.6
The definitive BASIC source
BASIC Training Guide
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

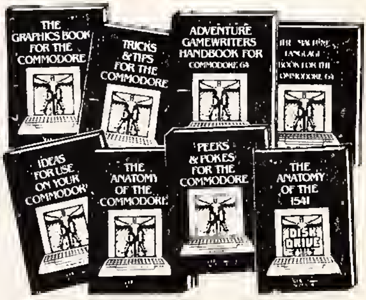
Introduction to programing; problem analysis; thorough description of all BASIC commands with hundreds of examples; monitor commends; utilities; much more. $16.95


COMMODORE 128 Vol.7
Useful programming quick hitters
PEEKS & POKES
A DATA - BECKER BOOK PUBLISHED BY
Abacus Software

Presents dozens of programming quick-hitters. Easy and useful techniques on the operating system, stacks, zero-page, pointers, the BASIC interpreter and more. $16.95



Essential guide for everyone interested in CP/M on the 128. Simple explanation of the operating system, memory usage, CP/M utility programs, submit files & more. $19.95


THE GRAPHICS BOOK FOR THE COMMODORE
TRICKS & TIPS FOR THE COMMODORE
ADVENTURE GAMEWRITERS HANDBOOK FOR COMMODORE 64
THE MACHINE LANGUAGE BOOK FOR THE COMMODORE 64
IDEAS FOR USE ON YOUR COMMODORE
THE ANATOMY OF THE COMMODORE
PEEKS & POKES FOR THE COMMODORE
THE ANATOMY OF THE 1541
DISK DRIVE

ANATOMY OF C-64 Insider's guide to the '64 Internals. Graphics, sound, I/O, kernal, memory maps, more. Complete commented ROM listings. 300pp $19.95

ANATOMY OF 1541 DRIVE Best handbook on floppy explains all. Many examples and utilities. Fully commented 1541 ROM listings. 500pp $19.95

REVISED & EXPANDED

MACHINE LANGUAGE C-64 Learn 6510 code write fast programs. Many samples and listings for complete assembler, monitor, & simulator. 200pp $14.95

GRAPHICS BOOK C-64 - best reference covers basic and advanced graphics. Sprites, animation, Hires, Multicolor, lightpen, 3D-graphics, IRQ, CAD, projections, curves, more. 350pp $19.95

TRICKS & TIPS FOR C-64 Collection of easy-to-use techniques: advanced graphics, improved data input, enhanced BASIC, CP/M, more. 275pp $19.95

1541 REPAIR & MAINTENANCE Handbook describes the disk drive hardware. Includes schematics and techniques to keep 1541 running. 200pp $19.95

ADVANCEO MACHINE LANGUAGE Not covered elsewhere: - video controller, interrupts, timers, clocks, I/O, real time, extended BASIC, more. 210pp $14.95

PRINTER BOOK C-64/VIC-20 Understand Commodore, Epson-compatible printers and 1520 plotter. Packed: utilities; graphics dump; 3D-plot; commented MPS801 ROM listings, more. 330pp $19.95

SCIENCE/ENGINEERING ON C-64 In depth intro to computers in science. Topics: chemistry, physics, biology, astronomy, electronics, others. 350pp $19.95

CASSETTE BOOK C-64/VIC-20 Comprehensive guide; many sample programs. High speed operating system fast file loading and saving. 225pp $14.95

IOEAS FOR USE ON C-64 Themes: auto expenses, calculator, recipe file, stock lists, diet planner, window advertising, others. Includes listings. 200pp $12.95

COMPILER BOOK C-64/C-128 All you need to know about compilers: how they work; designing and writing your own; generating machine code. With working example compiler. 300pp $19.95

Adventure Gamewriter's Handbook Step-by-step guide to designing and writing your own adventure games. With automated adventure game generator. 200pp $14.95

PEEKS & POKES FOR THE C-64 Includes in-depth explanations of PEEK, POKE, USR, and other BASIC commands. Learn the "inside" tricks to get the most out of your '64. 200pp $14.95

Optional Oiskettes for books
For your convenience, the programs contained in each of our books are available on diskette to save you time entering them from your keyboard. Specify name of book when ordering. $14.95 each

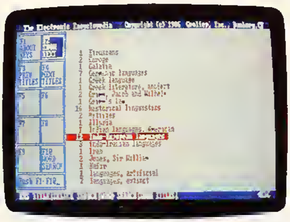
C-128 and C-64 are trademarks of Commodore Business Machines Inc.

Abacus Software

P.O. Box 7219 Dept. C9 Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Optional diskettes available for all book titles - $14.95 each. Other books & software also available. Call for the name of your nearest dealer. Or order directly from ABACUS using your MC, Visa or Amex card. Add $4.00 per order for shipping. Foreign orders add $10.00 per book. Call now or write for your free catalog. Dealer inquires welcome--over 1400 dealers nationwide.

the encyclopedia. If you're cross-referencing two words or phrases to see if they have any relationship to each other, you can choose from several Relation options. For instance, you can find out if your selected words or phrases appear in the same article, the same paragraph, within a certain number of words of each other, or in the exact order. The fifth option here, which can save you some time, lets you negate a word that might appear within the phrase you're looking for, but which is actually another subject entirely. If you are doing a report on Martin Luther, negating the word King will prevent you from pulling articles you don't need to read.



The Electronic Encyclopedia Copyright (c) 1986 Grolier, Inc., Danbury, CT

F1 ABOUT KEYS | F2 [illegible] | F3 PREV TITLES | F4 NEXT TITLES | F5 | F6 | F7 | F8 | F9 | F10 WORD SEARCH

Push F1-F10_

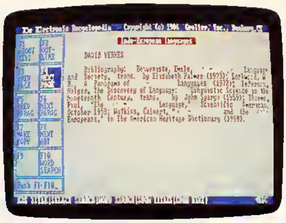
1 Etruscans
2 Europe
1 Galatia
7 Germanic languages
1 Greek language
1 Greek literature, ancient
2 Grimm, Jacob and Wilhelm
1 Grimm's law
16 historical linguistics
2 Hittites
1 Illyria
1 Indian languages, American
[illegible] Indo-European languages
3 Indo-Iranian languages
1 Iran
2 Jones, Sir William
1 Kafir
1 languages, artificial
7 languages, extinct

Step 2: After getting a list of entries, decide which you'd like to look at.

Let's say you're doing a research project on Indo-European culture. Upon entering that phrase, you'll find that there are 162 occurrences of that phrase in 65 articles. After asking to see a list of the articles, you can choose to read and even print out any of them. Moving around from article to article and in and out of searches is made quite simple by the function key menu that remains along the left side of the screen (and changes depending on what area of the software you're using).

To save you some time, if you don't want to skim through entire articles, every time your selected search word or phrase appears in an article or bibliography, it shows up as highlighted print.

The system's real power is quite evident the first time you sit down to conduct a search. The incredibly fast search capabilities were made possible by the software developers at Knowledge Set (formerly Activenture). In order to



The Electronic Encyclopedia Copyright (c) 1986 Grolier, Inc., Danbury, CT

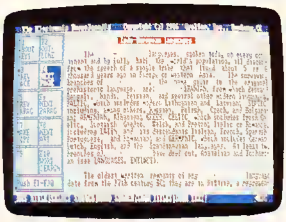
F1 ABOUT KEYS | F2 OUT-LINE | F3 PREV PAGE | F4 NEXT PAGE | F5 PREV PARAG | F6 NEXT PARAG | F7 MAKE COPY | F8 NEXT ART | F9 | F10 WORD SEARCH

Push F1-F10_

Indo-European languages

[illegible]

Bibliography: Benveniste, Emile, [illegible] Language and Society, trans. by Elizabeth Palmer (1973); Lockwood, W. B., A Panorama of [illegible] Languages (1972); Pedersen, Holger, The Discovery of Language: Linguistic Science in the Nineteenth Century, trans. by John Spargo (1959); Thieme, Paul, "The [illegible] Language," Scientific American, October 1958; Watkins, Calvert, "[illegible] and the [illegible]-Europeans," in The American Heritage Dictionary (1969).



Indo-European languages

The [illegible] languages, spoken today on every continent and by fully half the world's population, all descend from the speech of a single tribe that lived about 5 or 6 thousand years ago in Europe or western Asia. The surviving branches of [illegible], the nearest guide to the original prehistoric language, are: [illegible]-IRANIAN, from which descend Bengali, Hindi, Persian, and several other modern languages; BALTIC, which includes modern Lithuanian and Latvian; SLAVIC including, among others, Russian, Polish, Czech, and Bulgarian; ARMENIAN; Albanian; GREEK; CELTIC, which includes Irish Gaelic, Scottish Gaelic, Welsh, and Breton; Italic or Romance, including LATIN and its descendants Italian, French, Spanish, Portuguese, and Romanian; and GERMANIC, which includes German, Dutch, English, and the Scandinavian languages. At least ten branches of [illegible] have died out, Anatolian and Tocharian (see LANGUAGES, EXTINCT).

The oldest written remnants of any [illegible] language date from the 17th century BC; they are in Hittite, a recorded

Push F1-F10

The top screen shows (in highlighted text) where your selected phrase appears within a bibliography; the bottom screen shows it within an actual article about the topic. From here, you can print out a copy, continue your search, or begin a new search.

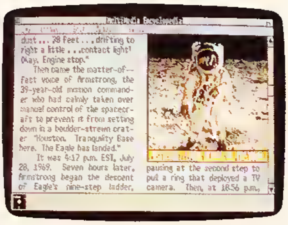
make referencing accurate and thorough, every unique word in the *Academic American Encyclopedia* was identified. Then the VAX minicomputer which compiled the list created an index that cross-referenced every entry. This accounts for the system's speed, as well as its ability to make connections between seemingly unrelated items that might never occur to the user, but which might make for some very interesting research.

Graphics And Sound, Too

Libraries and other institutions that have major information storage and retrieval needs have, understandably, shown a great deal of interest in CD-ROM. But there are still a few things that need to be worked out before CD-ROM becomes as commonplace as microfiche. First, compatibility: Ideally, CD-ROM should be a market similar to that of CD-audio; that is, any CD you buy will run on any manufacturer's CD-ROM player. Negotiations over standards are currently under way.

Second, where will the software come from? Many software publishers are very interested in developing for CD-ROM, though few have publicly committed to it. Part of the problem here stems from the old chicken-and-egg problem. Businesses are hesitant to buy a system unless there is a lot of software available, but software publishers are hesitant to put a lot of development money into a product unless there is a solid installed base of the hardware.

Sony and Philips recently announced specifications for a specialized kind of CD-ROM perhaps better suited to the home market. *CD-I (Compact Disc–Interactive)* suggests an environment that will allow the mixing of text, graphics, sound, and limited animation. It's described as a system, as opposed to CD-ROM, which is considered a peripheral. CD-I hardware may be available in several different configurations from several different companies, but the general idea is to get away from the need for any extensive technical knowledge to operate it. Several companies in the entertainment field have announced intentions to develop home entertainment products for the system.



Multimedia Encyclopedia

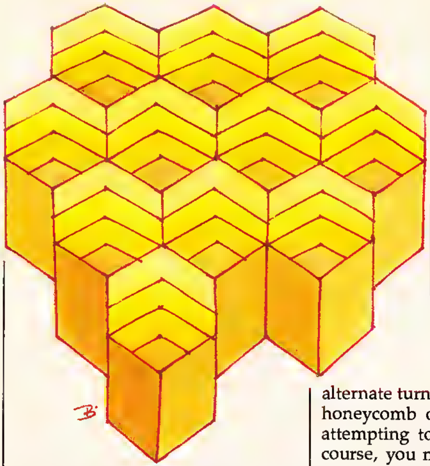
dust . . . 38 Feet . . . drifting to right a little . . . contact light! Okay. Engine stop."

Then came the matter-of-fact voice of Armstrong, the 39-year-old mission commander who had calmly taken over manual control of the spacecraft to prevent it from setting down in a boulder-strewn crater: "Houston, Tranquility Base here. The Eagle has landed."

It was 4:17 p.m. EST, July 28, 1969. Seven hours later, Armstrong began the descent of Eagle's nine-step ladder, pausing at the second step to pull a ring that deployed a TV camera. Then, at 10:56 p.m.,

Microsoft recently showed a prototype of the Multimedia Encyclopedia, *a CD-I product.*

Of course, better research tools won't necessarily mean better, smarter students. Motivation and the desire to learn are always key factors. But this new generation of electronic equipment will do much more than simply make it easier to find facts. Just as the computer age has so far sparked previously undreamed-of applications, so also may CD-ROM and CD-I technology lead to uses that we, at this early stage, can hardly imagine. ©



BEEHIVE

Steve Michel

To avoid getting stung in this delightful strategy game, you'll need to plan ahead. The original version of "Beehive" was written for the 512K Amiga. We've added fresh translations for the Commodore 64 (and 128 in 64 mode), Apple II series, IBM PC/PCjr, and Atari 400, 800, XL, and XE. The IBM PC/PCjr game requires a color/graphics card and BASICA for the PC, and Cartridge BASIC for the PCjr. The Atari version requires at least 32K of memory and a joystick. The Commodore version requires a joystick. The Apple II version requires a joystick and color monitor, and runs on any Apple II-series computer with either ProDOS or DOS 3.3.

"Beehive" is a two-player strategy game that requires you to concentrate fully and develop long-range planning skills. The game board consists of 121 hexagons arranged in a sloping 11 × 11 matrix. The name derives from the playing field's resemblance to the geometric precision of a honeycomb. The first player is assigned the left and right borders of the honeycomb, while the second player is assigned the top and bottom edges.

The object of the game is deceptively simple. Each player tries to connect a continuous line from one of his or her borders to the other. If you are player 1, for instance, you need to connect the left border with the right. The players alternate turns, filling in cells of the honeycomb one at a time. While attempting to complete your own course, you must also try to block your opponent's way, and this requires strategic thinking. The first player to connect both borders wins the game. As a reward, tiny bee faces appear along the line of connection, clearly marking the path to victory.

Entering The Game

Type in the program listing for your computer, referring to the special notes below. When you have saved a copy of the game, type RUN and press RETURN. Beehive begins by asking for the name of each player. After both players have entered their names, the beehive grid is drawn and play begins. In the Amiga and IBM PC/PCjr versions, the computer determines randomly which player should take the first turn; in other versions, player 1 always goes first. In the Amiga version, each player takes a turn by moving the mouse pointer to the desired cell and pressing the left mouse button once. Other versions substitute joystick or keyboard controls for the mouse (see below).

When you choose a cell, it is filled with a solid circle and your turn ends. While connecting your own borders, you should also be trying to prevent the other player from making a connection. Play continues until one player or the other completes a continuous line from one border to the other. At this point a victor is declared, and bee faces replace the circles along the entire winning route.

Winning Strategies

Like most two-player games, Beehive adjusts itself to the skill of the players. The basic concept is simple enough that even small children can enjoy playing. But when two knowledgeable players are matched, play proceeds at a much higher level. The flexibility of the game allows many different strategies.

Here are some important points for beginners to keep in mind. To begin with, your first move does *not* have to occur in one of your border rows. In fact, you can often establish a better strategic position by starting somewhere near the middle of the playing field. In a typical game you will have to swing back and forth between an expanding, offensive posture and a defensive, blocking posture. The middle areas accommodate both strategies well.

Second, it is not necessary that all of your cells be connected. That is, a new cell doesn't necessarily have to touch one of your existing cells. Any empty cell in the hive is fair game for either player, and it's often advantageous to space out your cells to allow multiple paths between borders. Starting multiple pathways makes it harder for an opponent to block your progress completely.

Finally, keep in mind that the hexagonal shape of each cell permits you to move in six different directions. Try not to get locked into a strict, straight-line strategy too often. Any pathway that connects both borders is legal, and in many cases the winning path will be quite roundabout.

Amiga Version

Before you begin typing in the Amiga version (Program 1), notice the small arrows marking the end of the line. They are not intended to be typed (in fact, we deliberately chose a character that's not available from the Amiga's keyboard). Instead, wherever you see an arrow in the listing, press RETURN or move the cursor off the line to enter it into memory.

The Amiga version of Beehive includes synthesized speech. Either player can toggle the speech effects on or off at any time. Press the left button once: A small box appears, indicating the current speech status. If speech was turned on, it is now turned off, and vice versa. Press the left button again to erase the speech box and resume the game.

Commodore 64/128 Version

The Commodore version (Program 2) runs on a Commodore 64 or Commodore 128 in 64 mode; it requires a joystick. Plug the joystick into port 1 and use it to move the bee-shaped pointer onto the desired cell. To select a cell, press the fire button.

Atari Version

Atari Beehive (Program 3) requires a joystick and runs on any Atari 400, 800, XL, or XE computer with at least 32K of memory. Plug the joystick into port 1. Move the pointer over the cell you wish to occupy, then press the fire button to select it.

Apple II Version

The Apple II version of Beehive (Program 4) runs on any Apple II-series computer, under DOS 3.3 or ProDOS. A color monitor and joystick are required. To select a cell, move the pointer onto it, then press the button.

IBM PC/PCjr Version

IBM Beehive (Program 5) requires a color/graphics card and BASICA for the IBM PC, and Cartridge BASIC for the PCjr. Keyboard controls are used to move the bee-shaped pointer around the playing field and to select a cell. Use the arrow keys to move left, right, up, or down. When the pointer is above the desired cell, press the space bar to select it.

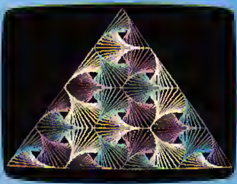
Program 1: Beehive For Amiga

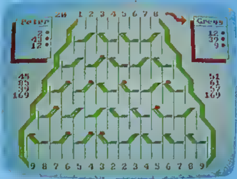
Please refer to the typing instructions in the article before entering this listing.

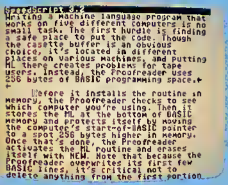
```
◂
CLS◂
talk$="": GOSUB talk◂
GOSUB init◂
GOSUB getnames ◂
start:◂
CLS: RANDOMIZE TIMER◂
markers = Ø: winner = Ø: prev.pl
ayer = Ø◂
player = INT(2*RND(1)+1)◂
FOR j = 1 TO 11: FOR k=1 TO 31:
hive%(j,k)=Ø: NEXT k: NEXT j ◂
FOR j = 1 TO 2Ø: pathlen(j) = Ø:
NEXT j◂
FOR j = 1 TO 65: path%(j) = Ø: u
sed%(j) = Ø: node%(j) = Ø: NEXT
j◂
GOSUB drawscreen◂
BREAK ON: ON BREAK GOSUB closeup
◂
◂
main:◂
IF prev.player <> player THEN◂
COLOR 4◂
LOCATE 1,2: PRINT "Player:
"◂
LOCATE 1,2: PRINT "Player: ";◂
COLOR colr(player): PRINT LEFT$(
player$(player),15)    ◂
talk$=player$(player): GOSUB tal
k◂
prev.player = player◂
END IF◂
WHILE MOUSE(Ø) = Ø◂
x = MOUSE(Ø)◂
a$=INKEY$:IF a$=" " THEN GOSUB r
eadkey◂
WEND◂
GOSUB checkmouse◂
IF used THEN main◂
GOSUB checkline◂
IF possible = 1 THEN GOSUB check
winner◂
LOCATE 3,2: PRINT "             "
◂
IF winner = 1 THEN drawpath◂
IF player = 1 THEN ◂
player = 2    ◂
ELSE   ◂
player = 1     ◂
END IF◂
GOTO main◂
◂
init:◂
CLS: colr(1) = 2: colr(2) = 3◂
DIM colcor%(11): FOR j = 1 TO 11
: READ colcor%(j): NEXT j◂
DATA 5,4,4,3,3,2,2,1,1,Ø,Ø◂
DIM row.inc%(6), col.inc%(6)◂
FOR j = 1 TO 6: READ row.inc%(j)
, col.inc%(j): NEXT j◂
DATA -1,-1,Ø,1,1,1,1,Ø,Ø,-1,-1,-
1◂
DIM hive%(11,31)◂
DIM used%(65), node%(65), path%(
65), pathlen(2Ø)◂
SCREEN 1,64Ø,2ØØ,3,2◂
WINDOW 1,"BEE HIVE",,16,1◂
GOSUB setcolor◂
DIM hexa(1ØØ),ball1(1ØØ),ball2(1
ØØ),eyes1(1ØØ),eyes2(1ØØ)◂
LINE (3Ø,1Ø)-(12,15),7: LINE - S
TEP (Ø,1Ø),7: LINE - STEP (18,5)
,7◂
LINE - STEP (18,-5),7: LINE - ST
EP (Ø,-1Ø),7: LINE - STEP (-18,-
5),7◂
LINE (3Ø,11)-(13,15),6: LINE - S
TEP (Ø,9),6: LINE - STEP (17,5),
6◂
LINE - STEP (16,-4),6: LINE - ST
EP (Ø,-1Ø),6: LINE - STEP (-17,-
4),6◂
GET (12,1Ø)-(48,3Ø),hexa ◂
CLS: CIRCLE (3Ø,2Ø),11,colr(1):
PAINT (3Ø,2Ø),colr(1): GET (2Ø,9
)-(4Ø,31),ball1◂
GOSUB parts: GET (18,12)-(42,3Ø)
, eyes1 ◂
CLS: CIRCLE (3Ø,2Ø),11,colr(2):
PAINT (3Ø,2Ø),colr(2): GET (2Ø,9
)-(4Ø,31),ball2◂
GOSUB parts: GET (18,12)-(42,3Ø)
, eyes2: CLS◂
RETURN◂
◂
parts:  ◂
CIRCLE (25,19),4,1: CIRCLE (35,1
9),4,1◂
PAINT (25,19),1: PAINT (35,19),1
◂
PSET (29,17): LINE - STEP (-5,-5
): LINE - STEP (-5,3)◂
PSET (31,17): LINE - STEP (5,-5)
: LINE - STEP (5,3)◂
CIRCLE (3Ø,24),2,1: PAINT (3Ø,24
),1◂
RETURN◂
◂
getnames:◂
COLOR 4◂
CLS: talk$="WELCOME TO BEEE HIVE
": GOSUB talk◂
a$ = " What is the name of playe
r 1 ": PRINT ◂
PRINT a$;: talk$=a$: GOSUB talk:
INPUT player$(1)◂
a$ = " What is the name of playe
r 2 ": PRINT ◂
PRINT a$;: talk$=a$: GOSUB talk:
INPUT player$(2)◂
talk$="Press space bar to turn s
peech off or on during game."◂
LOCATE 15,14:PRINT talk$◂
GOSUB talk:CLS: RETURN◂
◂
drawscreen:◂
CLS: y = 7◂
FOR r = 1 TO 11◂
x = 18Ø - r * 18◂
FOR c = 1 TO 11◂
x = x + 36◂
PUT (x,y),hexa,OR◂
NEXT c ◂
y = y + 15 ◂
NEXT r   ◂
PSET (595,12),2: GOSUB upndown:
LINE -STEP (Ø,1Ø),2◂
PSET (596,12),2: GOSUB upndown:
LINE -STEP (Ø,1Ø),2◂
PSET (597,12),2: GOSUB upndown:
LINE -STEP (Ø,1Ø),2◂
PSET (194,12),2: GOSUB upndown:
LINE -STEP (Ø,1Ø),2◂
PSET (195,12),2: GOSUB upndown:
LINE -STEP (Ø,1Ø),2◂
PSET (196,12),2: GOSUB upndown:
LINE -STEP (Ø,1Ø),2◂
y1=-5: y2=5: PSET (198,9),3: GOS
UB across◂
PSET (198,1Ø),3: GOSUB across◂
PSET (199,11),3: GOSUB across◂
y1=5: y2=-5: PSET (19,173),3: GO
SUB across◂
PSET (19,174),3: GOSUB across◂
PSET (19,175),3: GOSUB across◂
RETURN◂
◂
upndown:◂
FOR j = 1 TO 1Ø◂
LINE -STEP (Ø,1Ø),colr(1)◂
LINE -STEP (-18,5),colr(1)◂
NEXT j◂
RETURN◂
◂
```

GET UP TO 200 FUN-FILLED PROGRAMS EACH YEAR—

when you subscribe now to COMPUTE!







```
SpeedScript 3.2
Writing a machine language program that
works on five different computers is no
small task. The first hurdle is finding
a safe place to put the code. Though
the casette buffer is an obvious
choice, it's located in different
places on various machines, and putting
ML there creates problems for tape
users. Instead, the Proofreader uses
256 bytes of BASIC programming space.←
←
     Before it installs the routine in
memory, the Proofreader checks to see
which computer you're using. Then it
stores the ML at the bottom of BASIC
memory and protects itself by moving
the computer's start-of-BASIC pointer
to a spot 256 bytes higher in memory.
Once that's done, the Proofreader
activates the ML routine and erases
itself with NEW. Note that because the
Proofreader overwrites its first few
BASIC lines, it's critical not to
delete anything from the first portion
```

Subscribe to COMPUTE! today through this special introductory money-saving offer, and you'll be getting a lot more than just another computer magazine. That's because each issue of COMPUTE! comes complete with up to 20 all-new, action-packed programs.

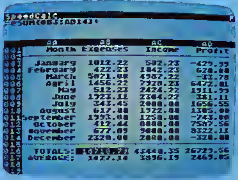
Subscribe now and you can depend on a steady supply of high quality, fun-filled programs like Hickory Dickory Dock, Switchbox, TurboDisk, Home Financial Calculator, Turbo Tape, SpeedScript, SpeedCalc, and hundreds of other educational, home finance, and game programs the entire family can use all year long.

The superb programs you'll find in each issue are worth much, much more than the low subscription price.

And there's more to COMPUTE! than just exciting new programs. Month after month, COMPUTE!'s superb articles deliver the latest inside word on everything from languages to interfaces...from programming to disk drives.

Whether you're a novice or an experienced user, COMPUTE! is the magazine for you. So subscribe today. Return the enclosed card or call 1-800-247-5470 (in Iowa, 1-800-532-1272).

<u>Do it now.</u>



```
SpeedCalc
=SUM(OB3:AO14)+

      AA        AB          AC          AD
    Month  Expenses      Income      Profit

  January   1012.22      582.23     -429.99
 February    422.23     1042.23      620.00
    March   5021.00     4987.22      -33.78
    April   1456.22     7928.23     6472.01
      May    512.23     2424.22     1911.99
     June   1923.11     2304.22      381.11
     July    343.45     2000.00     1656.55
   August    617.00     1922.00     1305.00
September   1993.00     1250.00     -743.00
  October    412.44     8000.00     7587.56
 November    677.89     9000.00     8322.11
 December   2320.00     2000.00     -320.00

  TOTALS: [illegible]   43440.35    26729.56
 AVERAGE:   1427.14     3896.19     2463.85
```

ACT NOW AND SAVE!

A Turning Point For Atari?
Report From The Winter Consumer Electronics Show
COMPUTE!
The Leading Magazine Of Home, Educational, And Recreational Computing
New Technologies: The Converging Digital Universe
Why Everything From Computers & CD-ROMs To TVs & Satellites Are On A Collision Course
Adding Power To ST BASIC
Valuable Techniques For Atari Programmers
Apple Hi-Res Graphics
Mix, Flip, And Copy High-Resolution Screens
64 Screen Genie
A Magic Method For Character Graphics
AmigaDOS Batch Files
Take Control Of Your Amiga
Atari FontMaker
Design Your Own Custom Characters

COMPUTE!™ Publications, Inc. abc
One of the ABC Publishing Companies

If attached order card is missing, write: COMPUTE!'s Gazette P.O. Box 10955, Des Moines, IA 50950

```
across:◂
FOR j = 1 TO 11◂
LINE -STEP (18,y1),colr(2)◂
LINE -STEP (18,y2),colr(2)◂
NEXT j◂
RETURN◂
◂
checkmouse:        ◂
x = MOUSE(3): y = MOUSE(4)◂
offset = Ø: used = Ø◂
yr = INT (y/15+.5): row = yr: yr
= yr * 15 ◂
IF INT (yr/2) = yr/2 THEN offset
= 18◂
xr = INT ((x-offset)/36+.5): col
= xr: xr = xr * 36 + offset◂
IF row < 1 OR row > 11 THEN◂
used = 1◂
RETURN◂
END IF◂
col = col - colcor%(row)◂
IF  col < 1 OR col > 11 THEN◂
used = 1◂
RETURN◂
END IF◂
rowhive = row: colhive = 1Ø+2*co
l-row◂
IF hive%(row,colhive) <> Ø THEN
◂
used = 1◂
RETURN◂
END IF  ◂
markers = markers + 1◂
hive%(row,colhive) = player◂
IF player = 1 THEN ◂
PUT (xr-1Ø,yr-9),ball1,OR ◂
ELSE   ◂
PUT (xr-1Ø,yr-9),ball2,OR◂
END IF◂
RETURN◂
◂
checkline:◂
possible=1◂
IF player = 2 THEN ◂
FOR row = 1 TO 6: ff=Ø: fb=Ø◂
FOR col = 1 TO 11: colhive=1Ø+2*
col-row◂
IF hive%(row,colhive)=player THE
N ff=1◂
colhive = 1Ø+2*(col)-(12-row)◂
IF hive%(12-row,colhive)=player
THEN fb=1◂
NEXT col  ◂
IF ff=Ø OR fb=Ø THEN ◂
possible = Ø◂
row = 1E+Ø9◂
END IF◂
NEXT row◂
ELSE◂
FOR col = 1 TO 6: ff=Ø: fb=Ø◂
FOR row = 1 TO 11: colhive=1Ø+2*
col-row◂
IF hive%(row,colhive)=player THE
N ff=1◂
colhive = 1Ø+2*(12-col)-row◂
IF hive%(row,colhive)=player THE
N fb=1◂
NEXT row ◂
IF ff=Ø OR fb=Ø THEN ◂
possible = Ø◂
col = 1E+Ø9◂
END IF◂
NEXT col◂
END IF◂
RETURN◂
◂
checkwinner:◂
LOCATE 3,2: COLOR 4: PRINT "Chec
king..."◂
used.cntr = Ø: winner = Ø: node.
cntr = Ø: node.total = Ø: counte
r = Ø◂
IF player = 1 THEN check1◂
FOR col = 1 TO 11: row = 1◂
IF hive%(row,1Ø+2*col-row) <> pl
ayer THEN skip2◂
noderow = row: nodecol = col: GO
SUB usedlookup◂
IF used.flag = 1 THEN skip2◂
node.total = 1: path.total = 1:c
ounter = 1◂
path%(1) = 1ØØ * noderow + nodec
ol◂
GOSUB checkpath◂
IF winner = 1 THEN col = 1E+Ø9◂
skip2:◂
NEXT col◂
RETURN◂
◂
check1:  ◂
FOR row = 1 TO 11: col = 1◂
IF hive%(row,1Ø+2*col-row) <> pl
ayer THEN skip1◂
noderow = row: nodecol = col: GO
SUB usedlookup◂
IF used.flag = 1 THEN skip1◂
node.total = 1: path.total = 1:
counter = 1◂
path%(1) = 1ØØ * noderow + nodec
ol◂
GOSUB checkpath◂
IF winner = 1 THEN row = 1E+Ø9◂
skip1:◂
NEXT row◂
RETURN◂
◂
usedlookup:◂
used.flag = Ø: search = 1ØØ * no
derow + nodecol◂
lk = Ø: IF used.cntr = Ø THEN sk
ipsearch◂
FOR lk = 1 TO used.cntr◂
IF search = used%(lk) THEN ◂
used.flag = 1◂
lk = 1E+Ø9◂
END IF  ◂
NEXT lk◂
skipsearch:◂
IF used.flag = Ø THEN◂
used.cntr = used.cntr + 1◂
used%(used.cntr) = search◂
END IF◂
RETURN◂
◂
checkpath:◂
node.cntr = Ø◂
FOR nc = 1 TO 6◂
noderow = noderow + row.inc%(nc)
: nodecol = nodecol + col.inc%(n
c)◂
IF noderow < 1 OR noderow > 11 O
R nodecol < 1 OR nodecol > 11 TH
EN skipnode
◂
IF hive%(noderow,1Ø+2*nodecol-no
derow) <> player THEN skipnode◂
GOSUB usedlookup: IF used.flag =
1 THEN skipnode◂
node.cntr = node.cntr + 1◂
node.total = node.total + 1: nod
e%(node.total) = 1ØØ * noderow +
nodecol◂
IF (player = 2 AND noderow = 11)
OR (player = 1 AND nodecol = 11)
THEN ◂
winner = 1◂
path.total = path.total + 1◂
path%(path.total) = 1ØØ * nodero
w + nodecol ◂
nc = 1E+Ø9◂
END IF  ◂
skipnode:◂
NEXT nc◂
IF winner = 1 THEN RETURN◂
IF node.cntr = Ø  AND node.total
= Ø THEN RETURN◂
IF node.cntr = Ø THEN ◂
path.total = path.total - pathle
n(counter)◂
pathlen(counter) = Ø◂
counter = counter - 1◂
END IF   ◂
IF node.cntr > 1 THEN counter =
counter + node.cntr - 1◂
noderow = INT(node%(node.total)/
1ØØ)◂
nodecol = node%(node.total) - 1Ø
Ø * noderow◂
path.total = path.total + 1◂
pathlen(counter) = pathlen(count
er) + 1◂
path%(path.total) = node%(node.t
otal)◂
node.total = node.total - 1 ◂
GOTO checkpath  ◂
◂
drawpath:◂
LOCATE 1,1: PRINT "
": COLOR 4◂
LOCATE 1,1: PRINT "THE WINNER: "
;:COLOR colr(player): PRINT play
er$(player);◂
a$ = "THE WINNER IS " + player$(
player): talk$=a$: GOSUB talk◂
FOR j = 1 TO path.total: offset
= Ø◂
row = INT(path%(j)/1ØØ): col = p
ath%(j) - 1ØØ*row + colcor%(row)
◂
IF row/2 = INT(row/2) THEN offse
t = 18 ◂
xr = col * 36 + offset: yr = row
* 15◂
IF player = 1 THEN◂
PUT (xr-1Ø,yr-9), ball1, XOR ◂
PUT (xr-12,yr-5), eyes1, OR◂
ELSE◂
PUT (xr-1Ø,yr-9), ball2, XOR◂
PUT (xr-12,yr-5), eyes2, OR◂
END IF◂
NEXT j  ◂
◂
goagain:◂
LINE (419,139)-(625,186),7,b: LI
NE (42Ø,14Ø)-(624,185),7,b◂
LINE (421,141)-(623,184),4,bf: C
OLOR 6◂
LOCATE 19,55: a$ = " WANT TO PLA
Y AGAIN ? ": PRINT  a$; ◂
LINE (431,162)-(487,18Ø),7,bf: L
OCATE 22,56: PRINT " YES ";◂
LINE (567,162)-(615,18Ø),7,bf: L
OCATE 22,73: PRINT " NO "; ◂
talk$=a$: GOSUB talk◂
◂
waiter:◂
WHILE MOUSE(Ø) <> 1◂
WEND◂
x = MOUSE(3): y = MOUSE(4)◂
IF y < 162 OR y > 18Ø THEN waite
r◂
IF x > 43Ø AND x < 488 THEN star
t◂
IF x > 566 AND x < 616 THEN clos
eup◂
GOTO waiter◂
◂
setcolor:◂
PALETTE Ø,.3,.3,.3     'grey◂
PALETTE 1,Ø,Ø,Ø        'black◂
PALETTE 2,Ø,1,Ø        'green◂
PALETTE 3,Ø,Ø,1        'blue◂
PALETTE 4,1,1,1        'white◂
PALETTE 5,Ø,1,1        'aqua◂
PALETTE 6,1,1,Ø        'yellow◂
PALETTE 7,.8,.2,Ø      'red◂
RETURN◂
◂
closeup:◂
PALETTE Ø,.1,.1,1      'blue◂
PALETTE 1,1,1,1        'white◂
PALETTE 2,Ø,Ø,Ø        'black◂
PALETTE 3,.85,.2,Ø     'red◂
WINDOW CLOSE 1◂
SCREEN CLOSE 1◂
```

Famous National Brand

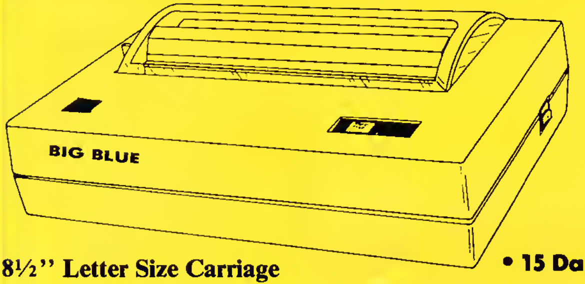
Commodore® • Atari® • Apple® • IBM®

Sale $39.95 List $199.00

Sale $39.95 List $199.00

8½" Letter Size 80 Column Printer Sale

• Word Processing • Program Listings • Graphics • Quiet Operation • Upper and Lower case • All points addressable Graphics • Underline • Enhanced • Much much More



8½" Letter Size Carriage

Super Quality

This printer was made by **Canon**® for one of the largest computer manufacturers in the world. The **Big Blue** printer comes ready to hook up to the serial port of the **IBM**®**PC jr.** Plus with low cost adapter cables you can connect the **Big Blue** printer to the **IBM**®**PC, IBM**®**XT, IBM**®**AT, Apple**®**II,IIe,IIc, Commodore**® **64,128, Atari**®, plus many more.

• 15 Day Free Trial

• 90 Day Immediate Replacement Warranty

Now you can have a full fledged 8½" letter size 80 column printer for less than the cost of a large box of paper. This printer uses advanced thermal technology to print upper and lower case (with true lower descenders), underline, enhanced, all points addressable graphics (works with Printshop) plus More. Print out pictures, program listings, wordprocessing pages, graphics and more. Perfect for the homeowner or student and definitely affordable. Fantastic Printer at an unbeatable price. List $199.00 **Sale $39.95**

Intelligent Commodore Interface— Allows you to connect the Big Blue printer to the printer port of the Commodore 64 and 128 computer. Print Commodore graphics, use Printshop, Word processors and more... List $49.95 **Sale $24.95**

Intelligent Atari Interface— Allows you to connect the Big Blue printer to Atari computers (except 1200). Print Atari graphics, Printshop ,word processors, and more... List $49.95 **Sale $24.95**

IBM®, Apple® RS-232 Adapter— Adapts the Big Blue printer to be used with any RS-232 port. List $19.95 **Sale $9.95**

Paper (2 Rolls) List $19.95 **Sale $9.95**

IBM, Apple, Canon, Commodore, Atari are trademarks of International Business Machines, Apple Computer, Canon Inc, Commodore Business Machines, Atari Inc. Respectively.

Add $10.00 for shipping, handling, and insurance. Illinois residents please add 6¼ % sales tax. Add 20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only.

PROTECTO
We Love Our Customers
22292 N. Pepper Rd., Barrington, Illinois 60010
312/382-5244 to order

COMMODORE 64 COMPUTER

(Order Now)

$149^95

- C128 Disks 79¢ ea.*
- Comstar Writer 64 $19.95
- 13" Color Monitor $139.95

COMMODORE 64 SYSTEM SALE

Commodore 64

Com. 1541 Disk Drive

13" Color Monitor

Plus $30.00 S&H

$457

(Ltd. Qty)

C128 Commodore Computer & 1571 Disk Drive

$499^00

- Voice Synthesizer $39.95
- 12" Monitor $79.95

CALL BEFORE YOU ORDER

SPECIAL SOFTWARE COUPON

PRICES MAY BE LOWER

COMMODORE 64 COMPUTER $149.95

You pay only $149.95 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your computer that allows you to SAVE OVER $250 off software sale prices!! With only $100 of savings applied, your net computer cost is $49.95!!

* C128 DOUBLE SIDED DISKS 79¢ EA.

Get these 5¼" Double Sided Floppy Disks specially designed for the Commodore 128 Computer (1571 Disk Drive). 100% Certified, ***Lifetime Warranty***, Automatic Lint Cleaning Liner included. 1 Box of 10 - $9.90 (99¢ ea.), 5 Boxes of 10 - $44.50 (89¢ ea.), 10 Boxes of 10 -$79.00 (79¢ ea.).

13" COLOR MONITOR $139.95

You pay only $139.95 when you order this 13" COLOR MONITOR. LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your monitor that allows you to save over $250 off software sale prices!! With only $100 of savings applied, your net color monitor cost is only $39.95. (16 Colors).

Premium Quality 150-170 CPS Comstar Aero 160 Printer $199.00

The COMSTAR Aero 160 gives you a 10" carriage, 150-170 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin setting, true lower decenders with super and subscripts, prints standard, block graphics and special characters. It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) List $499.00 **Sale $199.00.**

9" SAMSUNG GREEN SCEEEN MONITOR

Super High Resolution composite green screen monitor. Perfect for 80 column use with The C128 computer (Req. $19.95 Cable)
List $129.95 **Sale $59.95.**

80 COLUMNS IN COLOR COMSTAE WRITEE 64 WORD PROCESSOR $19.95

This COMSTAR WRITER 64 WORD PROCESSOR is the finest avaliable for the COMMODORE 64 computer! The ULTIMATE FOR PROFESSIONAL Word Processing, DISPLAYS 40 or 80 COLUMNS IN COLOR or black and white! Simple to operate, powerful text editing, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! List $99.00. **SALE $19.95.** ***Ltd. Qty Closeout Item***

We pack a SPECIAL SOFTWAEE DISCOUNT COUPON with every COMMODOEE 64 COMPUTEE, DISK DRIVE, PRINTEE, or MONITOR we sefil This coupon allows you to SAVE OVEE $250 OFF SALE PRICES!!

(Examples)

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
PaperClip	$59.95	$34.95	$29.95
Consultant	$59.95	$49.95	$39.95
Leader Board	$39.95	$24.95	$22.95
The Print Shop	$44.95	$27.95	$26.95
Halley's Project	$39.95	$22.95	$19.95
Practicalc (spread sheet)	$59.95	$19.95	$14.95
Voice Command Module	$79.95	$39.95	$34.95
Nine Princes in Amber	$32.95	$24.95	$21.95
Super Bowl Sunday	$35.00	$22.95	$19.95
Flip and File Disk Filer	$24.95	$14.95	$12.95
Pro Joy Stick	$19.95	$12.95	$10.00
PartyWare	$19.95	$14.95	$11.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60
Financial Planner Sylvia Porter	$59.95	$38.95	$35.95
Hardball	$29.95	$18.95	$16.95
C64 Troubleshoot & Repair Guide	$24.95	$15.95	$12.95

(See over 100 coupon items in our catalog)

Write or call for Sample SPECIAL SOFTWARE COUPON!

ATTENTION Computer Clubs

We Offer Big Volume Discounts

CALL TODAY!

PROTECTO WARRANTY

All Protecto's products carry a minimum 90 day warranty. If anything fails within 90 days from the date of purchase, simply send your product to us via United Parcel Service prepaid. We will IMMEDIATELY send you a replacement at no charge via United Parcel Service prepaid. This warranty proves once again that ***We Love Our Customers.***

C128 COMPUTER & 1571 $499.00

Now you can get the C128 Commodore computer & the 1571 Disk Drive for one low price of only $499.00. List $698.00
SALE $499.00.

340K 1571 COMMODORE DISK DRIVE $259.00

Double Sided, Single Disk Drive for C-128 allows you to use C-128 mode plus CPM mode. 17 times faster than 1541, plus runs all 1541 formats.
List $349.00. **Sale $259.00.**

SUPER AUTO DIAL MODEM $29.95

Easy to use. Just plug into your Commodore 64 computer and you're ready to transmit and receive messages. Easier to use than dialing your telephone, just push one key on your computer! Includes exclusive easy to use program for up and down loading to printer and disk drives.
Best in U.S.A. List $99.00. **SALE $29.95.**
Coupon $24.95.

VOICE SYNTHESIZER $39.95

For Commodore-64 computers. Just plug it in and you can program words and sentences, adjust volume and pitch, make talking adventure games, sound action games and customized talkies!! PLUS ($19.95 value) TEXT TO SPEECH program included FREE, just type a word and hear your computer talk — ADD SOUND TO "ZORK", SCOTT ADAMS AND OTHER ADVENTURE GAMES!! (Disk or tape.) List $89.00. **SALE $39.95**

12" SAMSUNG AMBER SCREEN MONITOR $79.95

Super High Resolution amber screen monitor. 80 columns x 24 lines, easy to read, a must for wordprocessing . Fantastic value.
List $129.00. **Sale $79.95.**
(C128 cable $19.95. C64, Atari cable $9.95)

PRINTER/TYPEWRITER COMBINATION $199.95

Superb letter quality, daisy wheel printer/typewriter combination. Two machines in one — just a flick of the switch. Extra large carriage, typewriter keyboard, automatic margin control compact,lightweight, drop in cassette ribbon! (90 day warranty) centronics parallel interface Built -in. List $349.00. **SALE $199.95.**
(Ltd. Qty.)

14" RGB & COMPOSITE COLOR MONITOR $239.95

Must be used to get 80 columns in color with 80 column computers (C128 - IBM - Apple). (RGB Cable $19.95) Add $14.50 shipping.
List $399.00. **SALE $239.95.**

• LOWEST PRICES • 15 DAY FREE TRIAL
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL

PHONE ORDERS
8 a.m. - 8 p.m. C.S.T. Weekdays
9 a.m. - 12 noon C.S.T. Saturdays

• 90 DAY FREE REPLACEMENT WARRANTY
• OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6¼% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Prices & Availability subject to change without notice.
VISA — MASTER CARD — C.O.D. No. C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010
312/382-5244 to order

Famous Comstar National Brand

10″ Printer Sale

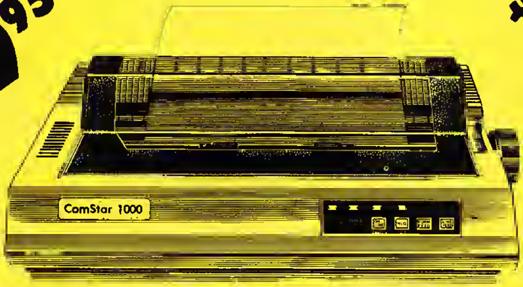
Includes Commodore

Near Letter Quality **Interface** *Near Letter Quality*

Best Value in the USA

• 100 CPS draft/20CPS near-letter quality • Dot Addressable Graphics • Adjustable Tractor and Friction Feed • Automatic Paper Loading • Right and Left Margin settings • Pica, Elite, Condensed, Italics • Superscript • Subscript • Underline, Bold print, Double Strike • Superb NEAR LETTER QUALITY

$179⁹⁵



$179⁹⁵

Easy to Use

Fantastic Graphics

2 Year Warranty

The Comstar 1000 is one of the best values in the United States today. Print your letters, documents, programs, pictures, and more at a blazing 100 Characters Per Second or 20 cps in the Near Letter quality mode. (Looks just like it came from a typewriter.) Plus, choose your printing mode (NLQ, Draft) from your wordprocessor or by simply pushing the controls on the front panel. Fantastic Quality at a Fantastic Price.List $349.00 SALE$179.95.

Print Method
Serial impact dot matrix (9 pin)
Print Speed
Draft- 100 CPS NLQ- 20 CPS
Character Sets
96 ASCII Characters, Marker, Symbols (Includes Italic font)

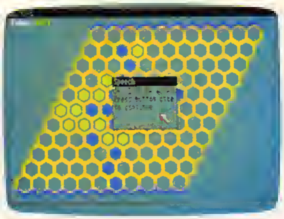
Ribbon (Life exp.)
Black: cassette (2.5 million characters)
Dimensions
15.4 (W) x 10.9 (D) x 4.7 (H) inch
Weight
Approx. 10 lbs
Character Spacing
Fixed

Line Spacing
1/6, 1/8, 7/72, and 1/216 inch
Paper Feed
Adjustable tractor and friction feed
Paper feeding Direction
Bi-directional
Copies
2 plus original

Supply is Limited so Buy Today

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
22292 N. Pepper Rd., Barrington, Illinois 60010
312/382-5244 to order



"Beehive" for the 512K Amiga, a challenging strategy game.

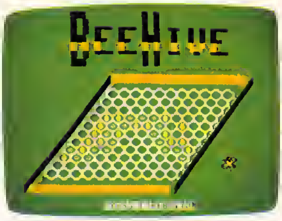
```
STOP◂
◂
◂
readkey:◂
WINDOW 4,"Speech",(250,70)-(390,
110),16,1◂
IF TalkFlag=1 THEN◂
talk$="Now I can talk."◂
PRINT talk$◂
TalkFlag=1-TalkFlag◂
GOSUB talk◂
GOTO clearmouse◂
END IF◂
IF TalkFlag=Ø THEN◂
talk$="OK, I'll be quiet."◂
PRINT talk$◂
GOSUB talk◂
TalkFlag=1-TalkFlag◂
END IF◂
◂
clearmouse:◂
WHILE MOUSE(Ø)<>Ø:WEND◂
PRINT "Press button once"◂
PRINT "to continue..."◂
WHILE MOUSE(Ø)<>1:WEND◂
WHILE INKEY$<>"":WENO◂
WINDOW CLOSE 4◂
RETURN◂
◂
talk:◂
IF TalkFlag=Ø THEN SAY TRANSLATE
$(talk$)◂
RETURN◂
◂
```

Program 2: Commodore 64/128 Beehive

Version by Kevin Mykytyn, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
EF 10  POKE56,4B:CLR:GOTO8Ø
RK 2Ø  GOSUB56Ø
OK 3Ø  JV=15-(PEEK(56321)AND15)
       -128*((PEEK(56321)AND16)
       <>16):IFJV>127THENRETURN
AX 4Ø  IFJV=ØTHEN3Ø
FF 5Ø  TX=X:TY=Y:TX=TX+X(JV):TY
       =TY+Y(JV)
HK 6Ø  IFTX<1ORTX>11ORTY<1ORTY>
       11THEN3Ø
BO 7Ø  X=TX:Y=TY:GOSUB56Ø:GOTO3
       Ø
XO BØ  GOSUB17Ø:GOSUB53Ø
RM 9Ø  POKE532BØ,5:POKE53281,5:
       PRINT"{CLR}{6 DOWN}
       {6 RIGHT}";:GOSUB47Ø:POK
       E53269,1
SC 1ØØ  INPUT"{BLK}{3 DOWN}
        {2 RIGHT}ENTER YOUR NAM
        E PLAYER ONE";PN$(1)
SR 11Ø  INPUT"{DOWN}{2 RIGHT}EN
        TER YOUR NAME PLAYER TW
        O";PN$(2):POKE53269,Ø
```

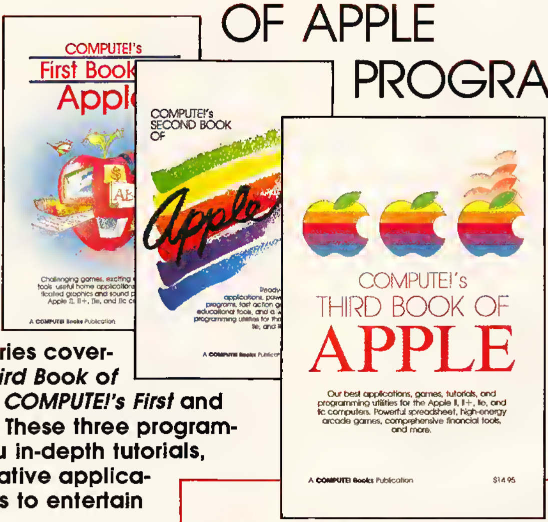


The Commodore 64/128 version of "Beehive" features a bee-shaped pointer.

```
OC 12Ø GOSUB4ØØ:FORA=1TO2:PN$(
       A)=LEFT$(PN$(A),15):NEX
       T:X=1:P=1:UN=1:Y=1
HM 13Ø B$="YOUR TURN ":GOSUB11
       6Ø
RM 14Ø GOSUB2Ø:SP=1397+4Ø*Y+X*
       2-Y
MM 15Ø IFPEEK(SP)<>32THENF=1Ø:
       GOSUB58Ø:GOTO14Ø
AK 16Ø BD(X,Y)=UN:POKESP,81:PO
       KESP+54272,7*(P-1):GOSU
       B59Ø:GOSUB61Ø:P=3-P:GOT
       O13Ø
KF 17Ø ML$="【I】"+CHR$(8)+"【X】<
       "+CHR$(3)+"【2】XJ"+CHR$(
       16)+CHR$(248)+"L【B】【T】"
       :POKE835,Ø
SK 18Ø POKE836,2ØB:POKE83Ø,Ø:P
       OKE831,216:POKE828,Ø:PO
       KE829,56:POKE56334,Ø
SP 19Ø POKE1,51:ML$=ML$:SYS(PE
       EK(51)+256*PEEK(52)):PO
       KE1,55:POKE56334,1
GF 2ØØ FORI=12568TO12631:READJ
       :POKEI,J:NEXT:POKE53272
       ,28
AP 21Ø FORA=ØTO1Ø:READX(A),Y(A
       ):NEXTA:FORA=832TO895:R
       EADB:POKEA,B:NEXT
JE 22Ø POKE53276,1:POKE2Ø4Ø,13
       :POKE53287,7:POKE53285,
       Ø:RETURN
ER 23Ø OATA231,126,24,24,24,24
       ,126,231
RC 24Ø OATAØ,Ø,Ø,Ø,Ø,Ø,126,231
SC 25Ø DATA231,126,Ø,Ø,Ø,Ø,Ø,Ø
HR 26Ø DATA7,3Ø,24,24,24,24,12
       6,231
OA 27Ø DATA7,3Ø,24,24,24,24,3Ø
       ,7
MF 28Ø DATA224,12Ø,24,24,24,24
       ,12Ø,224
RA 29Ø DATA231,126,24,24,24,24
       ,12Ø,224
JA 3ØØ DATA195,36,126,219,255,
       126,36,24
PH 31Ø DATA Ø,Ø,Ø,-1,Ø,1,Ø,Ø,-
       1,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1,Ø,Ø,Ø
       ,Ø,Ø
CO 32Ø DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,16
CS 33Ø DATAØ,Ø,65,BØ,Ø,65,164,
       2Ø
JR 34Ø OATA7Ø,1ØØ,1,15Ø,1ØØ,1,
       165,144
JM 35Ø DATAØ,1Ø6,64,5,1Ø5,Ø,26
       ,17Ø
GB 36Ø OATA64,21,153,144,26,86
       ,8Ø,5
SE 37Ø DATA5,144,Ø,2,96,Ø,1,16
       Ø
SH 3BØ DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø
JG 39Ø OATAØ,Ø,Ø,Ø,Ø,Ø,Ø,53
GH 4ØØ PRINT"{CLR}{8 DOWN}"SPC
       (13)"{RVS}{BLK}£{YEL}
       【J】{23 SPACES}{BLK}£"
CB 41Ø PRINTSPC(12)"{BLK}{RVS}
       £{OFF}£";:FORA=1TO11:
       PRINT"{WHT}$ ";:NEXTA:P
       RINT" {BLK}{RVS}£{OFF}
       £"
BC 42Ø FORA=1TO11
OA 43Ø PRINTSPC(12-A)"{BLK}
       {RVS}£{OFF}£{WHT}& ";
       :FORB=1TO1Ø:PRINT"# ";:
       NEXT:PRINT"){RVS}{BLK}
       £{OFF}£":NEXTA
GP 44Ø PRINT"{BLK}{RVS}£{OFF}
       £{2 SPACES}";:FORA=1TO
       11:PRINT"{WHT}% ";:NEXT
       :PRINT"{LEFT}{BLK}{RVS}
       £{OFF}£"
FM 45Ø PRINT"{BLK}£{YEL}{RVS}
       {23 SPACES}{OFF}【H】
       {BLK}£":PRINT"{HOME}
       {7 SPACES}";
XJ 46Ø POKE1827,39:POKE56Ø99,1
       :POKE1459,4Ø:POKE55731,
       1
GH 47Ø PRINT"{BLK}{RVS}【K】

       {2 SPACES}{OFF}
       {9 SPACES}{RVS}【K】{OFF}
       【K】{RVS}【K】{OFF}【K】
       {BLK}"SPC(24)"{RVS}【K】
       {OFF}【K】{RVS}【K】{OFF}
       【K】"SPC(8)"{RVS}【K】
       {OFF}【K】{RVS}【K】{OFF}
       【K】"SPC(24);
MJ 48Ø PRINT"{RVS}【K】{OFF}【K】
       {RVS}【K】{OFF}【K】{RVS}
       【K】【D】【I】{OFF}【V】{RVS}
       【K】【O】【I】{OFF}【V】{RVS}
       【K】{OFF}【K】{RVS}【K】
       {OFF}【K】 {RVS}【F】【D】
       {OFF} {RVS}【K】{OFF}【K】
       {RVS}【K】{OFF}【K】{RVS}
       【K】【O】【I】{OFF}【V】"SPC(1
       2)"{RVS}【K】{2 SPACES}
       {OFF} {RVS}【K】【C】";
PP 49Ø PRINT"{OFF}【F】 {RVS}【K】
       【C】{OFF}【F】 {RVS}【K】
       {2 SPACES}{OFF}【K】
       {RVS}【K】{OFF}【K】 {RVS}
       【K】{OFF}【K】{RVS}【K】
       {OFF}【K】{RVS}【K】【C】
       {OFF}【F】{YEL}"SPC(13)"
       {RVS}【K】{OFF}【K】{RVS}
       【K】{OFF}【K】{RVS}【K】
       {OFF}【K】{2 SPACES}{RVS}
       【K】{OFF}【K】{2 SPACES}
       {RVS}【K】";
HO 5ØØ PRINT"{OFF}【K】{RVS}【K】
       {OFF}【K】 {RVS}【K】{OFF}
       【K】 【C】{RVS}【C】【V】{OFF}
       【V】{RVS}【K】{OFF}【K】
       {BLK}"SPC(14)"{RVS}【K】
       {OFF}【K】{RVS}【K】{OFF}
       【K】【C】{RVS}【2 I】{OFF}
       【V】【C】{RVS}【2 I】{OFF}
       【V】{RVS}【K】{OFF}【K】
       {RVS}【K】{OFF}【K】 {RVS}
       【2 I】{OFF} ";
FC 51Ø PRINT" 【C】【V】 【C】{RVS}
       【2 I】{OFF}【V】"SPC(12)"
       {RVS}【K】{2 SPACES}{OFF}
       "SPC(9)"{RVS}【K】{OFF}
       【K】{RVS}【K】{OFF}【K】":
SH 52Ø POKE5324B,3Ø:POKE53264,
       1:POKE53249,15Ø:RETURN
JF 53Ø FORA=54272TO54295:POKEA
       ,Ø:NEXT:POKE54296,15:PO
       KE54277,25
MX 54Ø OIMTA(11,11,2),EH(61),E
       V(61)
SE 55Ø DIMBD(11,11),SH(5Ø),SV(
       5Ø):SP$="{RVS}":FORA=1T
       O2Ø:SP$=SP$+" ":NEXT:RE
       TURN
KM 56Ø POKE53249,Ø:TX=X*16+(11
       -Y)*B+36:POKE53248,TXAN
       D255
```

Another Fantastic Collection OF APPLE PROGRAMS


COMPUTE!'s
First Book of
Apple
Challenging games, exciting tools, useful home applications, sophisticated graphics and sound for the Apple II, II+, IIe, and IIc computers
A COMPUTE! Books Publication
COMPUTE!'s
SECOND BOOK
OF
Apple
Ready-to-type applications, powerful programs, fast action games, educational tools, and a wealth of programming utilities for the Apple II+, IIe, and IIc
A COMPUTE! Books Publication
COMPUTE!'s
THIRD BOOK OF
APPLE
Our best applications, games, tutorials, and programming utilities for the Apple II, II+, IIe, and IIc computers. Powerful spreadsheet, high-energy arcade games, comprehensive financial tools, and more.
A COMPUTE! Books Publication
$14.95

COMPUTE! Books continues the tradition of excellent Apple-II series coverage with *COMPUTE!'s Third Book of Apple*, the follow-up to *COMPUTE!'s First* and *Second Book of Apple*. These three program-packed books offer you in-depth tutorials, valuable utilities, innovative applications, and terrific games to entertain the whole family.

COMPUTE!'s First Book of Apple
Edited
228 pages
A collection of 35 exciting games, educational programs, home applications, and graphics routines for owners of Apple II, IIe, and IIc computers.
$12.95, ISBN 0-942386-69-8

COMPUTE!'s Second Book of Apple
Edited
271 pages
Over 30 arcade-style games, applications, educational and logic games, graphics generators, and programming utilities for the Apple II+, IIe, and IIc. *A disk is also available for $12.95 which includes programs in the book, 0084BDSK.*
$12.95, ISBN 0-87455-008-4

COMPUTE!'s Third Book of Apple
Edited
The best Apple applications, games, tutorials, and utilities from *COMPUTE!* magazine and *COMPUTE!'s Apple Applications Special* have been brought together for the third volume in this bestselling series. Understandable tutorials help you get the most out of your Apple II, II+, IIe, or IIc personal computer. Ready-to-type-in games—from vivid, exciting arcade games such as "High Rise" to thought-provoking strategy games like "Switchbox"—make the Apple an entertainment delight. Practical applications, like "Apple *SpeedCalc*," an impressive spreadsheet program, and "Your Personal Ledger," a complete financial tool, put the power in your Apple. Also included are utilities that will make programming easier and faster. All programs have been thoroughly tested and are ready to type in. Error-checking utilities insure that each program is typed in correctly the first time. *A disk is available for $12.95 which includes all the programs in the book, 637BDSK.*
$14.95, ISBN 0-87455-063-7

Your whole family will enjoy and use the business, educational, and recreational programs and tutorials in our three books of Apple.

Order one, two, or all three books today.

Call toll-free 800-346-6767 (in NY 212-887-8525), or write COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

All orders must be prepaid in U.S. funds. NC residents add 4.5 percent sales tax. NY residents add 8.25 percent sales tax. Please include $2.00 shipping per book for U.S. and surface mail; $5.00 airmail.

Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

```
PS 570  POKE53264,-(TX>255):POK
        E53249,Y*8+120:POKE5326
        9,1:RETURN
BM 580  POKE54273,F:POKE54276,1
        6:POKE54276,17:RETURN
XQ 590  POKE54273,10:POKE54276,
        64:POKE54276,65:FORZZ=1
        5TO1STEP-.3:POKE54275,Z
        Z:NEXT
HP 600  RETURN
PG 610  CH=X:CV=Y:LC=Ø:RC=Ø:FOR
        X=-1TO1:TH=CH+X
JK 620  TV=CV-1-(X=1):GOSUB78Ø
JG 630  TV=CV-(X>-1):GOSUB78Ø
BK 640  NEXT
PM 650  IFP=1ANDCH=1ORP=2ANDCV=
        1THENLC=1
EK 660  IFP=1ANDCH=11ORP=2ANDCV
        =11THENRC=2
QJ 670  CC=LC+RC:IFCC=3THENBD(C
        H,CV)=UN+1:GOTO85Ø
JA 680  IFCC=ØTHEN77Ø
BS 690  SP=Ø:SH(Ø)=CH:SV(Ø)=CV
AM 700  IFSP=-1THEN77Ø
RC 710  DH=SH(SP):DV=SV(SP):SP=
        SP-1
CH 720  BD(DH,DV)=UN+CC
EP 730  FORX=-1TO1:TH=DH+X
FJ 740  TV=DV-1-(X=1):GOSUB82Ø
JH 750  TV=DV-(X>-1):GOSUB82Ø
MC 760  NEXT:GOTO7ØØ
ED 770  X=CH:Y=CV:UN=5-UN:RETUR
        N
DD 780  IFTH<1ORTH>11ORTV<1ORTV
        >11THENRETURN
DP 790  IFBD(TH,TV)=UN+1THENLC=
        1
PS 800  IFBD(TH,TV)=UN+2THENRC=
        2
SK 810  RETURN
SA 820  IFTH<1ORTH>11ORTV<1ORTV
        >11THENRETURN
FP 830  IFBD(TH,TV)=UNTHENSP=SP
        +1:SH(SP)=TH:SV(SP)=TV
CM 840  RETURN
PM 850  POKE53248,33:POKE53264,
        1:POKE53249,195
PS 860  FORZZ=1TO2Ø:POKE646,ZZ:
        B$=" YOU WIN ":GOSUB117
        Ø:NEXT:GOSUB116Ø
CQ 870  FORCC=1TO2:FE=1:LE=1:EH
        (1)=CH:EV(1)=CV:EF=Ø:L=
        1
HA 880  CD=CH:IFP=2THENCD=CV
KF 890  IFCC=1ANDCD=1ORCC=2ANDC
        D=11THENHH(CC)=Ø:GOTO99
        Ø
HB 900  NE=LE:E=FE
RA 910  DH=EH(E):DV=EV(E)
JQ 920  FORX=-1TO1:TH=DH+X:TV=D
        V-1-(X=1):GOSUB1Ø9Ø:TV=
        DV-(X>-1):GOSUB1Ø9Ø:NEX
        T
GM 930  IFEF=1THEN99Ø
XS 940  IF(E=LE)THEN97Ø
EK 950  E=E+1:IFE=61THENE=1
XA 960  GOTO91Ø
CM 970  FE=LE+1:LE=NE:IFFE=61TH
        ENFE=1
EC 980  L=L+1:GOTO9ØØ
AE 990  NEXT:FORCC=1TO2:DH=HH(C
        C):DV=VV(CC):L=TA(DH,DV
        ,CC):IFDH=ØTHEN1Ø4Ø
BF 1000  POKE781,DV+9:POKE782,D
         H*2-DV+13:POKE783,Ø:SY
         S6552Ø:PRINT"{OFF}{6}*
         ":GOSUB59Ø
ED 1010  IFL=1THEN1Ø4Ø
QQ 1020  FORX=-1TO1:TH=DH+X:TV=
         DV-1-(X=1):GOSUB1Ø6Ø:T
         V=DV-(X>-1):GOSUB1Ø6Ø:
         NEXT
KM 1030  L=L-1:DH=AH:DV=AV:GOTO
         1ØØØ
EA 1040  NEXT:POKE781,CV+9:POKE
         782,CH*2-CV+13:POKE783
         ,Ø:SYS6552Ø:PRINT"{6}*
         ":GOSUB59Ø
HH 1050  GOTO119Ø
AC 1060  IFTH<1ORTH>11ORTV<1ORT
         V>11THENRETURN
MF 1070  IFTA(TH,TV,CC)=L-1THEN
         AH=TH:AV=TV
XF 1080  RETURN
KB 1090  IFTH<1ORTH>11ORTV<1ORT
         V>11THENRETURN
QJ 1100  IFBD(TH,TV)<>UN+CCORTA
         (TH,TV,CC)<>ØTHENRETUR
         N
FB 1110  TA(TH,TV,CC)=L:NE=NE+1
         :IFNE=61THENNE=1
FF 1120  EH(NE)=TH:EV(NE)=TV
SD 1130  CD=TH:IFP=2THENCD=TV
DP 1140  IFCC=1ANDCD=1ORCC=2AND
         CD=11THENEF=1:HH(CC)=T
         H:VV(CC)=TV
SK 1150  RETURN
KH 1160  POKE646,7*(P-1)
FR 1170  POKE214,23:PRINT:A$=LE
         FT$(SP$,(16-LEN(PN$(P)
         )/2))+B$+PN$(P)
JA 1180  PRINT" "A$;LEFT$(SP$,4
         Ø-LEN(A$));:RETURN
CD 1190  POKE214,23:PRINT:PRINT
         "{WHT}{13 SPACES}{RVS}
         PRESS FIREBUTTON{OFF}
         {1Ø SPACES}";
PE 1200  WAIT56321,16,16:POKE21
         4,23:PRINT:PRINTSPC(12
         )"{OFF}{2Ø SPACES}";
AR 1210  FORA=1TO11:FORB=1TO11:
         FORC=1TO2:TA(A,B,C)=Ø:
         BD(A,B)=Ø
XA 1220  NEXTC,B,A:POKE53269,Ø:
         GOTO12Ø
```

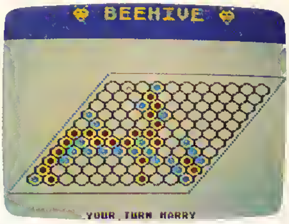
Program 3: Atari Beehive

Version by Kevin Mykytyn, Editorial Programmer

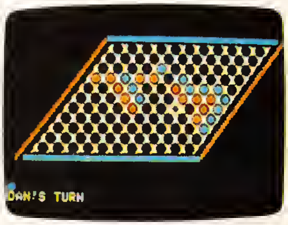
For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
BJ 10  POKE 1Ø6,96:GOSUB 2ØØØ
       :GOTO BØ
LB 20  FL=Ø:GOSUB 56Ø
NK 30  JV=15-STICK(Ø)+12B*(ST
       RIG(Ø)=Ø):IF JV>127 TH
       EN RETURN
JC 40  IF JV=Ø THEN 3Ø
NP 50  POKE 77,Ø:TX=X:TY=Y:TX
       =TX+X(JV):TY=TY+Y(JV)
DB 60  IF TX<1 OR TX>11 OR TY
       <1 OR TY>11 THEN 3Ø
IJ 70  X=TX:Y=TY:FL=Ø:GOSUB 5
       6Ø:GOTO 3Ø
FK B0  GOSUB 53Ø:GOSUB 17Ø:PR
       INT "{CLEAR}"
DC 90  DIM T$(3Ø),TM$(3Ø),NAM
       E$(4Ø),LENGTH(2):FOR A
       =1 TO 4Ø:NAME$(A,A)="
       ":NEXT A:POSITION 17,1
       Ø:PRINT "BEEHIVE"
HE 100  FOR A=1 TO 2:PRINT "
        {2 DOWN}ENTER YOUR NA
        ME PLAYER ";A;" ";:IN
        PUT T$
EC 101  IF T$="" THEN T$=" "
OB 105  IF LEN(T$)>15 THEN T$
        =T$(1,15)
NA 106  LENGTH(A)=LEN(T$)
EC 110  NAME$((A-1)*15+1,(A-1
        )*15+LEN(T$))=T$:NEXT
         A:DY=1
DL 120  GOSUB 4ØØ:X=1:P=1:UN=
        1:Y=1
EB 140  T$="YOUR TURN ":GOSUB
         4ØØØ:POKE 712,15Ø-98
        *(P=2)
AB 145  GOSUB 2Ø:LOCATE X*2-Y
        +14,Y+5,SP:POSITION X
        *2-Y+14,Y+5:PRINT CHR
        $(SP)
LO 150  IF SP<>32 THEN SOUND
        1,1ØØ,12,15:FOR TO=1
        TO 5Ø:NEXT TO:SOUND 1
        ,Ø,Ø,Ø:GOTO 14Ø
NL 160  BD(X,Y)=UN:POSITION X
        *2-Y+14,Y+5:PRINT CHR
        $(42+P):GOSUB 59Ø
CP 162  GOSUB 61Ø:P=3-P:GOTO
        14Ø
EC 170  FOR A=Ø TO 1Ø23:POKE
        24576+A,PEEK(57344+A)
        :NEXT A
BN 175  FOR A=256ØØ TO 25B56:
        POKE A,Ø:NEXT A
CP 180  FOR I=246ØØ TO 247Ø3:
        READ J:POKE I,J:NEXT
        I
IN 210  FOR A=Ø TO 1Ø:READ X,
        Y:X(A)=X:Y(A)=Y:NEXT
        A
KP 220  POKE 54279,64:POKE 53
        277,3:POKE 559,62:POK
        E 623,1:POKE 7Ø4,Ø:RE
        TURN
NJ 230  DATA 231,126,24,24,24
        ,24,126,231
DD 240  DATA Ø,Ø,Ø,Ø,Ø,Ø,126,
        231
DE 250  DATA 231,126,Ø,Ø,Ø,Ø,
        Ø,Ø
EH 260  DATA 7,3Ø,24,24,24,24
        ,126,231
LO 270  DATA 7,3Ø,24,24,24,24
        ,3Ø,7
NB 2B0  DATA 224,12Ø,24,24,24
        ,24,12Ø,224
NL 290  DATA 231,126,24,24,24
        ,24,12Ø,224
CC 300  DATA 195,36,126,219,2
        55,126,36,24
BK 301  DATA Ø,4Ø,17Ø,17Ø,17Ø
        ,17Ø,4Ø,Ø
LL 302  DATA Ø,2Ø,B5,85,85,B5
        ,2Ø,Ø
LI 303  DATA 2,2,B,B,32,32,12
        B,128
KC 304  DATA B5,Ø,Ø,Ø,Ø,Ø,Ø,Ø
KO 305  DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,B5
MI 310  DATA Ø,Ø,Ø,-1,Ø,1,Ø,Ø
        ,-1,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1,Ø
        ,Ø,Ø,Ø,Ø
KE 400  GOSUB 2ØØØ:POKE 756,9
        6:POKE 752,1:DL=PEEK(
        56Ø)+256*PEEK(561):PO
        KE DL+6,7:POKE DL+3,7
        1
NO 405  POSITION 5,Ø:PRINT "*
         BEEHIVE *":POSITION
        15,4:PRINT "/////////
        /////////////////"
FF 410  PRINT SPC$(1,12);"-";
        :FOR A=1 TO 11:PRINT
        "$ ";:NEXT A:PRINT "
        -"
DB 420  FOR A=1 TO 11
NO 430  PRINT SPC$(1,12-A);"-
        & ";:FOR B=1 TO 1Ø:PR
        INT "# ";:NEXT B:PRIN
        T ")-":NEXT A
CB 440  PRINT "-  ";:FOR A=1
        TO 11:PRINT "% ";:NEX
        T A:PRINT "{LEFT}-"
CJ 450  PRINT "..............
        ..........":POSITION
        4,16:PRINT "'":POSITI
        ON 36,6:PRINT "(":RET
        URN
DH 530  DIM TA(11,35),EH(61),
        EV(61),BD(11,11),SH(5
        Ø),SV(5Ø),SP$(2Ø),X(1
```

```
         Ø),Y(1Ø),8PC$(2Ø),HH(
         15),VV(15)
BC 535   PDKE 752,1:POSITION 1
         4,1Ø:PRINT "PLEASE WA
         IT"
PH 54Ø   GOSUB 3ØØØ:FDR A=1 TD
          2Ø:8P$(A,A)="-":SPC$
         (A,A)=" ":NEXT A:RETU
         RN
EM 56Ø   L8=PEEK(88):H8=PEEK(8
         9):POKE 752,1:POKE 89
         ,1ØØ:PDKE 88,DY*8+99:
         POSITIDN Ø,Ø:PRINT "
         {8 SPACES}":IF FL THE
         N 575
IO 57Ø   POKE 53248,X*8+(11-Y)
         *4+61:POKE 88,Y*8+99:
         POSITION Ø,Ø:PRINT "⊠
         {■}▦, &# "
BM 575   POKE 88,L8:POKE 89,H8
         :DY=Y:RETURN
DP 59Ø   FDR T=15 TO Ø STEP -Ø
         .4:SOUNO 1,1ØØ,1Ø,T:N
         EXT T:RETURN
JL 61Ø   CH=X:CV=Y:LC=Ø:RC=Ø:F
         DR X=-1 TO 1:TH=CH+X
BB 62Ø   TV=CV-1+(X=1):GDSUB 7
         8Ø
OC 63Ø   TV=CV+(X>-1):GOSUB 78
         Ø
OB 64Ø   NEXT X
BK 65Ø   IF P=1 ANO CH=1 OR P=
         2 AND CV=1 THEN LC=1
IE 66Ø   IF P=1 ANO CH=11 DR P
         =2 ANO CV=11 THEN RC=
         2
BA 67Ø   CC=LC+RC:IF CC=3 THEN
          8D(CH,CV)=UN+1:GOTD
         85Ø
ON 68Ø   IF CC=Ø THEN 77Ø
AH 69Ø   SP=Ø:SH(Ø)=CH:SV(Ø)=C
         V
DB 7ØØ   IF SP=-1 THEN 77Ø
LJ 71Ø   OH=SH(SP):OV=SV(8P):S
         P=SP-1
FO 72Ø   8D(OH,OV)=UN+CC
BK 73Ø   FDR X=-1 TO 1:TH=OH+X
BA 74Ø   TV=OV-1+(X=1):GOSUB 8
         2Ø
OB 75Ø   TV=OV+(X>-1):GOSUB 82
         Ø
OO 76Ø   NEXT X:GOTD 7ØØ
GA 77Ø   X=CH:Y=CV:UN=5-UN:RET
         URN
MB 78Ø   IF TH<1 OR TH>11 OR T
         V<1 OR TV>11 THEN RET
         URN
OA 79Ø   IF 8D(TH,TV)=UN+1 THE
         N LC=1
DA 8ØØ   IF 8D(TH,TV)=UN+2 THE
         N RC=2
HJ 81Ø   RETURN
MB 82Ø   IF TH<1 OR TH>11 OR T
         V<1 OR TV>11 THEN RET
         URN
MB 83Ø   IF 8D(TH,TV)=UN THEN
         8P=SP+1:SH(8P)=TH:SV(
         8P)=TV
HM 84Ø   RETURN
GA 85Ø   FL=1:GDSUB 56Ø:T$="YD
         U WIN ":GDSUB 4ØØØ:FD
         R A=255 TD Ø STEP -1:
         PDKE 712,A:NEXT A
KA 86Ø   PDSITIDN Ø,2Ø:PRINT "
         {15 SPACE8}SEARCHING
         {13 SPACES}"
BP 87Ø   FOR CC=1 TD 2:FE=1:LE
         =1:EH(1)=CH:EV(1)=CV:
         EF=Ø:L=1
AO 88Ø   CO=CH:IF P=2 THEN CD=
         CV
JO 89Ø   IF CC=1 AND CD=1 OR C
         C=2 AND CO=11 THEN HH
         (CC)=Ø:GOTO 99Ø
EB 9ØØ   NE=LE:E=FE
MI 91Ø   OH=EH(E):OV=EV(E)
MK 92Ø   FOR X=-1 TO 1:TH=OH+X
         :TV=OV-1+(X=1):GDSUB
         1Ø9Ø:TV=OV+(X>-1):GOS
         UB 1Ø9Ø:NEXT X
PF 93Ø   IF EF=1 THEN 99Ø
FP 94Ø   IF (E=LE) THEN 97Ø
FF 95Ø   E=E+1:IF E=61 THEN E=
         1
HC 96Ø   GOTD 91Ø
BA 97Ø   FE=LE+1:LE=NE:IF FE=6
         1 THEN FE=1
NO 98Ø   L=L+1:GOTO 9ØØ
AO 99Ø   NEXT CC:FDR CC=1 TO 2
         :OH=HH(CC):OV=VV(CC):
         L=TA(OH,OV*3+CC):IF D
         H=Ø THEN 1Ø4Ø
CI 1ØØØ  POSITION OH*2-OV+14,
         OV+5:PRINT "*":GOSUB
          59Ø
PP 1Ø1Ø  IF L=1 THEN 1Ø4Ø
OM 1Ø2Ø  FOR X=-1 TO 1:TH=DH+
         X:TV=OV-1+(X=1):GOSU
         8 1Ø6Ø:TV=OV+(X>-1):
         GOSU8 1Ø6Ø:NEXT X
FP 1Ø3Ø  L=L-1:OH=AH:OV=AV:GO
         TO 1ØØØ
CI 1Ø4Ø  NEXT CC:POSITION CH*
         2-CV+14,CV+5:PRINT "
         *":GOSU8 59Ø
MK 1Ø5Ø  GOTO 116Ø
OD 1Ø6Ø  IF TH<1 OR TH>11 OR
         TV<1 OR TV>11 THEN R
         ETURN
ON 1Ø7Ø  IF TA(TH,TV*3+CC)=L-
         1 THEN AH=TH:AV=TV
KJ 1Ø8Ø  RETURN
PB 1Ø9Ø  IF TH<1 OR TH>11 OR
         TV<1 OR TV>11 THEN R
         ETURN
CI 11ØØ  IF 8O(TH,TV)<>UN+CC
         OR TA(TH,TV*3+CC)<>Ø
          THEN RETURN
NL 111Ø  TA(TH,TV*3+CC)=L:NE=
         NE+1:IF NE=61 THEN N
         E=1
KO 112Ø  EH(NE)=TH:EV(NE)=TV
EK 113Ø  CO=TH:IF P=2 THEN CO
         =TV
O6 114Ø  IF CC=1 ANO CO=1 DR
         CC=2 ANO CO=11 THEN
         EF=1:HH(CC)=TH:VV(CC
         )=TV
KH 115Ø  RETURN
JF 116Ø  POSITION Ø,2Ø:PRINT
         "{12 SPACES}PRESS FIR
         E8UTTDN{8 SPACES}"
AM 1165  IF STRIG(Ø)<>Ø THEN
         1165
IK 117Ø  POSITION Ø,2Ø:PRINT
         "{36 8PACES}":GOSU8 3
         ØØØ:8DTD 12Ø
BE 2ØØØ  GRAPHICS Ø:POKE 71Ø,
         15:POKE 7Ø9,Ø:PDKE 7
         Ø8,45:POKE 54279,96:
         POKE 559,62:PDKE 7Ø4
         ,1Ø2:RETURN
OI 3ØØØ  FOR A=1 TD 11:FDR 8=
         1 TO 35:TA(A,8)=Ø:NE
         XT 8:NEXT A:FDR A=1
         TD 11:FDR 8=1 TO 11:
         8D(A,8)=Ø:NEXT 8:NEX
         T A
KE 3Ø1Ø  RETURN
PL 4ØØØ  TM$=T$:TM$(LEN(T$)+1
         ,LEN(T$)+LENGTH(P))=
         NAME$((P-1)*15+1,(P-
         1)*15+LENGTH(P))
AJ 4ØØ5  POSITION 2,2Ø:PRINT
         "{33 8PACE8}"
FL 4Ø1Ø  PDSITION 19-LEN(TM$)
         /2,2Ø:PRINT TM$:RETU
         RN
```



"Beehive" for Atari 400, 800, XL, and XE computers.



Apple II version of "Beehive."

Program 4: Apple II Beehive

Version by Tim Victor, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
#0 1ØØ LDMEM: 16384: OIM 8D(11,1
       1),SH(5Ø),SV(5Ø),TA(11,11
       ,2),EH(61),EV(61)
6F 11Ø FDR A = 768 TO A + 88: RE
       AD D: POKE A,D: NEXT : RE
       AD D: IF D < > - 1 THEN 1
       Ø7Ø
A6 12Ø FOR A = 35328 TD A + 7: P
       DKE A,Ø: NEXT
3B 13Ø FOR A = 3584Ø TD A + 79:
       READ D: PDKE A,D: NEXT :
       READ D: IF D < > - 1 THEN
        1Ø7Ø
44 14Ø TEXT : HOME : FOR I = 1 T
       D 2
30 15Ø PRINT "PLAYER "I"'S NAME:
       ";: INPUT "";A$:NN$(I) =
       LEFT$ (A$,24): NEXT
B7 16Ø PDKE 6,Ø: PDKE 7,138: IF
       PEEK (19Ø * 256) < > 76 T
       HEN POKE 54,Ø: POKE 55,3:
        CALL 1ØØ2: GDTO 18Ø
B6 17Ø PRINT CHR$ (4);"PR#A$3ØØ"
59 18Ø GDSUB 85Ø
F0 19Ø P = 1:UN = 1:NH = 6:NV =
       6:CH = 6:CV = 6: GDSUB 93
       Ø
7A 2ØØ HTAB 1: VTAB 21: CALL - 8
       68: PRINT NN$(P)"'";:S =
       ASC ( RIGHT$ (NN$(P),1)):
        IF S - 32 * (S > 96) < >
        83 THEN PRINT "S";
C9 21Ø PRINT " TURN": VTAB 2Ø: H
       TAB 1: PRINT CHR$ (96 + P
       )
99 22Ø IF PEEK (49249) > 127 THE
       N 29Ø
A3 23Ø IF PDL (Ø) < 9Ø THEN NH =
        NH - 1: IF NH < 1 THEN N
       H = 1
23 24Ø IF PDL (Ø) > 165 THEN NH
       = NH + 1: IF NH > 11 THEN
        NH = 11
```

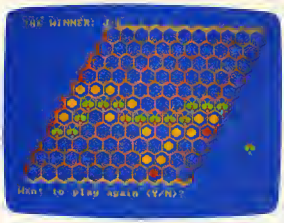
```
21 25Ø IF PDL (1) < 9Ø THEN NV =
        NV - 1: IF NV < 1 THEN N
       V = 1
9B 26Ø IF POL (1) > 165 THEN NV
       = NV + 1: IF NV > 11 THEN
        NV = 11
EC 27Ø IF CH < > NH OR CV < > NV
        THEN GOSUB 96Ø:CH = NH:C
       V = NV: GOSUB 93Ø
1E 28Ø GOTD 22Ø
FE 29Ø IF BD(CH,CV) < > Ø THEN P
       RINT CHR$ (7);: GOTO 23Ø
CB 3ØØ BD(CH,CV) = UN
07 31Ø GOSUB 96Ø: GDSUB 93Ø
BD 32Ø IF PEEK (49249) > 127 THE
       N 32Ø
D7 33Ø LC = Ø:RC = Ø: FOR X = -
       1 TD 1:TH = CH + X
E7 34Ø TV = CV - 1 + (X = 1): GO
       SUB 5ØØ
06 35Ø TV = CV + (X > - 1): GOSU
       B 5ØØ
07 36Ø NEXT
5B 37Ø IF P = 1 ANO CH = 1 OR P
       = 2 ANO CV = 1 THEN LC =
       1
9C 38Ø IF P = 1 ANO CH = 11 OR P
        = 2 AND CV = 11 THEN RC
       = 2
7E 39Ø CC = LC + RC: IF CC = 3 T
       HEN 57Ø
D1 4ØØ IF CC = Ø THEN 49Ø
68 41Ø SP = Ø:SH(Ø) = CH:SV(Ø) =
        CV
C0 42Ø IF SP = - 1 THEN 49Ø
2D 43Ø OH = SH(SP):OV = SV(SP):S
       P = 8P - 1
38 44Ø 8O(OH,DV) = UN + CC
C9 45Ø FOR X = - 1 TD 1:TH = OH
       + X
2F 46Ø TV = OV - 1 + (X = 1): GO
       8U8 54Ø
4C 47Ø TV = DV + (X > - 1): GOSU
       8 54Ø
70 48Ø NEXT : GOTO 42Ø
1D 49Ø P = 3 - P:UN = 5 - UN: GO
       TO 2ØØ
FE 5ØØ IF TH < 1 OR TH > 11 OR T
       V < 1 OR TV > 11 THEN RET
       URN
2A 51Ø IF 8D(TH,TV) = UN + 1 THE
       N LC = 1
37 52Ø IF BD(TH,TV) = UN + 2 THE
       N RC = 2
1B 53Ø RETURN
07 54Ø IF TH < 1 OR TH > 11 OR T
       V < 1 OR TV > 11 THEN RET
       URN
D5 55Ø IF BD(TH,TV) = UN THEN SP
        = SP + 1:SH(SP) = TH:SV(
       SP) = TV
21 56Ø RETURN
80 57Ø GOSU8 96Ø: VTA8 21: HTA8
       1: CALL - 86B: PRINT NN$(
       P)" WINS!": PRINT "CHECKI
       NG BOARO"
1E 58Ø FOR CC = 1 TO 2:FE = 1:LE
        = 1:EH(1) = CH:EV(1) = C
       V:EF = Ø:L = 1
C1 59Ø CO = CH: IF P = 2 THEN CD
        = CV
CD 6ØØ IF CC = 1 ANO CO = 1 OR C
       C = 2 ANO CO = 11 THEN HH
       (CC) = Ø: GOTD 7ØØ
4D 61Ø NE = LE:E = FE
D5 62Ø OH = EH(E):DV = EV(E)
D9 63Ø FOR X = - 1 TO 1:TH = OH
       + X:TV = OV - 1 + (X = 1)
       : GOSU8 78Ø:TV = OV + (X
       > - 1): GOSU8 78Ø: NEXT
FB 64Ø IF EF = 1 THEN 7ØØ
19 65Ø IF (E = LE) THEN 68Ø
5E 66Ø E = E + 1: IF E = 61 THEN
        E = 1
22 67Ø GOTO 62Ø
F1 68Ø FE = LE + 1:LE = NE: IF F
       E = 61 THEN FE = 1
55 69Ø L = L + 1: BOTO 61Ø
08 7ØØ NEXT : FOR CC = 1 TO 2:DH
        = HH(CC):DV = VV(CC):L =
        TA(DH,OV,CC): IF DH = Ø
       THEN 74Ø
80 71Ø HTAB OH * 2 - OV + 14: VT
       AB OV + 5: PRINT CHR$ (1Ø
       5);: IF L = 1 THEN 74Ø
C9 72Ø FOR X = - 1 TD 1:TH = OH
       + X:TV = OV - 1 + (X = 1)
       : GOSUB 75Ø:TV = OV + (X
       > - 1): GOSUB 75Ø: NEXT
3C 73Ø L = L - 1:OH = AH:OV = AV
       : GDTO 71Ø
1C 74Ø NEXT : HTA8 CH * 2 - CV +
        14: VTAB CV + 5: PRINT C
       HR$ (1Ø5);: GOSUB 1Ø1Ø: G
       DTO 1BØ
0B 75Ø IF TH < 1 OR TH > 11 OR T
       V < 1 OR TV > 11 THEN RET
       URN
DD 76Ø IF TA(TH,TV,CC) = L - 1 T
       HEN AH = TH:AV = TV
25 77Ø RETURN
11 78Ø IF TH < 1 OR TH > 11 OR T
       V < 1 OR TV > 11 THEN RET
       URN
9F 79Ø IF BO(TH,TV) < > UN + CC
       DR TA(TH,TV,CC) < > Ø THE
       N RETURN
76 8ØØ TA(TH,TV,CC) = L:NE = NE
       + 1: IF NE = 61 THEN NE =
        1
B9 81Ø EH(NE) = TH:EV(NE) = TV
1D 82Ø CO = TH: IF P = 2 THEN CO
        = TV
11 83Ø 1F CC = 1 ANO CO = 1 OR C
       C = 2 ANO CO = 11 THEN EF
        = 1:HH(CC) = TH:VV(CC) =
        TV
20 84Ø RETURN
F9 85Ø HGR : HOME : FOR I = 6 TO
        16: VTA8 I: HTA8 2Ø - I
95 86Ø FOR J = 1 TO 11: PRINT CH
       R$ (96); CHR$ (32);: NEXT
        : PRINT CHR$ (96);
B0 87Ø FOR J = Ø TO 1: HTAB 18 -
        I + J: PRINT CHR$ (99 +
       J + 2 * (I < > 2 * INT (I
        / 2)));: HTA8 43 - I + J
       : PRINT CHR$ (99 + J + 2
       * (I = 2 * INT (I / 2)));
       : NEXT
6A 88Ø NEXT : HCOLOR= 4: FOR I =
        Ø TD 4: HPLOT 92 + I,38
       TO 14 + I,127: HPLOT 255
       + I,3B TO 177 + I,127: NE
       XT
08 89Ø VTA8 5: HTAB 13: PRINT CH
       R$ (1Ø1);
E6 9ØØ FOR J = 1 TO 12: PRINT CH
       R$ (1Ø3); CHR$ (1Ø4);: NE
       XT : PRINT CHR$ (99);
22 91Ø VTA8 17: HTAB 2: PRINT CH
       R$ (1Ø2);
0E 92Ø FOR J = 1 TO 12: PR1NT CH
       R$ (1Ø4); CHR$ (1Ø3);: NE
       XT : PRINT CHR$ (1ØØ);: R
       ETURN
A8 93Ø HCOLOR= 7
F4 94Ø GV = CV * B + 32:GH = 92
       + 7 * (CH * 2 - CV)
8F 95Ø HPLOT GH,GV TO GH + 4,GV
       TO GH + 7,GV + 4 TO GH +
       4,GV + 7 TO GH,GV + 7 TD
       GH - 3,GV + 3 TO GH,GV: R
       ETURN
5D 96Ø HCOLOR= 4: GOSU8 94Ø
5A 97Ø VTA8 CV + 5: HTA8 14 + CH
        * 2 - CV:A$ = CHR$ (32)
F2 98Ø IF 8O(CH,CV) > 3 THEN A$
       = CHR$ (97 + (CV < > 2 *
       INT (CV / 2))): GOTO 1ØØØ
E2 99Ø IF BD(CH,CV) > Ø THEN A$
       = CHR$ (9B - (CV < > 2 *
       INT (CV / 2)))
FD 1ØØØ FRINT A$;: RETURN
85 1Ø1Ø VTAB 22: HTAB 1: PRINT "
        PRESS KEY TO QUIT, BUTTD
        N TD PLAY AGAIN"
ED 1Ø2Ø IF PEEK (49249) > 127 TH
        EN 1Ø5Ø
82 1Ø3Ø IF PEEK (49152) < 12B TH
        EN 1Ø2Ø
5F 1Ø4Ø POKE 49168,Ø: NORMAL : E
        ND
EA 1Ø5Ø HGR : FOR I = 1 TO 11: F
        DR J = 1 TD 11:BD(I,J) =
         Ø:TA(I,J,1) = Ø:TA(I,J,
        2) = Ø: NEXT : NEXT
E9 1Ø6Ø RETURN
2F 1Ø7Ø PRINT "ERROR IN OATA STA
        TEMENTS": ENO
82 1ØBØ OATA 216,12Ø,133,69,134,
        7Ø,132,71,166,7,1Ø
FA 1Ø9Ø OATA 1Ø,176,4,16,62,4B,4
        ,16,1,232,232
F6 11ØØ OATA 1Ø,134,27,24,1Ø1,6,
        133,26,144,2,23Ø
93 111Ø OATA 27,165,4Ø,133,B,165
        ,41,41,3,5,23Ø
8A 112Ø OATA 133,9,162,B,16Ø,Ø,1
        77,26,36,5Ø,4B
31 113Ø DATA 2,73,127,164,36,145
        ,B,23Ø,26,2ØB,2
C2 114Ø OATA 23Ø,27,165,9,24,1Ø5
        ,4,133,9,2Ø2,2ØB
69 115Ø OATA 226,165,69,166,7Ø,1
        64,71,88,76,24Ø,253
05 116Ø OATA 255
25 117Ø OATA -1
CE 118Ø OATA 193,182,156,156,156
        ,156,182,193,128,136,17Ø
0A 119Ø OATA 17Ø,17Ø,17Ø,136,128
        ,12B,148,213,213,213,213
FC 12ØØ OATA 148,128,128,128,192
        ,192,192,2ØB,2Ø8,14B,138
44 121Ø OATA 138,13Ø,13Ø,128,128
        ,12B,12B,128,128,128,128
EA 122Ø OATA 16Ø,16Ø,168,168,148
        ,133,133,129,129,129,12B
BB 123Ø OATA 128,128,128,17Ø,17Ø
        ,17Ø,17Ø,128,128,128,128
C7 124Ø OATA 213,213,213,213,128
        ,128,193,162,156,2Ø1,2Ø1
25 125Ø OATA 19Ø,156,136
23 126Ø OATA -1
```

Program 5: IBM PC/PCjr Beehive

Version by Patrick Parrish, Programming Supervisor

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
MI 1Ø KEY OFF:OEF SEG=Ø:POKE 1Ø4
      7,PEEK(1Ø47) OR 64:GOTD 2Ø
J0 2Ø GOSU8 35Ø
LC 3Ø GOSU8 65Ø
NP 4Ø REM START
QL 5Ø RANOOMIZE TIMER
H8 6Ø WINNER=Ø:PREV.PLAYER=Ø:ROW
      =6:COL=6:C8=146:R8=89
MC 7Ø PLAYER=INT(2*RNO+1)
J8 8Ø LOCATE 12,11:PRINT "Please
       wait a moment"
KJ 9Ø FOR J=1 TO 11:FOR K=1 TO 1
      1:HIVE%(J,K)=Ø:NEXT K:NEXT
       J
H6 1ØØ FOR J=1 TO 2Ø:PATHLEN(J)=
       Ø:NEXT J
AA 11Ø FOR J=1 TO 65:PATH%(J)=Ø:
       USEO%(J)=Ø:NDDE%(J)=Ø:NEX
       T J
II 12Ø GOSU8 7ØØ:LOCATE 24,1:PRI
       NT "Player:";
```



THE WINNER: J

Want to play again (Y/N)?

Keyboard controls are used in the IBM PC/PCjr version of "Beehive."

```
AE 130 REM MAIN
PC 140 IF PREV.PLAYER=PLAYER THE
       N 170
EP 150 LOCATE 24,B:PRINT "
                 ";:LOCATE 24,8:P
       RINT PLAYER$(PLAYER);:IF
       PLAYER=1 THEN PUT (270,13
       0),EYE81,PSET ELSE PUT (2
       70,130),EYES2,P8ET
MH 160 PREV.PLAYER=PLAYER
MN 170 PUT (CB,RB),BEE
DN 180 OEF SEG=0:POKE 1050,PEEK(
       1052)
DK 190 A$=RIGHT$(INKEY$,1):IF LE
       N(A$)=0 THEN 190
PD 200 PUT (CB,RB),BEE:OCOL=COL:
       OROW=ROW
NN 210 IF A$=CHR$(77) THEN ROW=R
       OW+1:COL=COL+1:IF ROW>11
       OR COL>11 THEN ROW=OROW:C
       OL=OCOL EL8E R8=R8+15:CB=
       CB+9
CB 220 IF A$=CHR$(75) THEN ROW=R
       OW-1:COL=COL-1:IF ROW<1 O
       R COL<1 THEN ROW=OROW:COL
       =OCOL ELSE R8=RB-15:CB=CB
       -9
CH 230 IF A$=CHR$(72) THEN ROW=R
       OW-1-(ROW=1):IF ROW<>OROW
        THEN CB=C8+9:R8=R8-15
HO 240 IF A$=CHR$(B0) THEN ROW=R
       OW+1+(ROW=11):IF ROW<>ORO
       W THEN`CB=C8-9:RB=RB+15
NK 250 PUT (CB,RB),BEE
PA 260 IF A$=" " THEN GOSUB 1050
        ELSE 180
BB 270 IF USEO THEN PUT (CB,RB),
       8EE:GOTO 140
OG 280 OROW=ROW:OCOL=COL:GOSUB 1
       110
NG 290 IF POSSI8LE=1 THEN GOSUB
       1290
IH 300 IF WINNER=1 THEN 1BB0
PO 310 LOCATE 24,1:PRINT "P1ayer
       :";
CC 320 IF PLAYER=1 THEN PLAYER=2
        ELSE PLAYER=1
BC 330 RDW=DROW:CDL=OCOL:GOTO 14
       0
LO 340 REM INIT
AC 350 CLS:COLR(1)=2:CDLR(2)=3
EA 360 OIM RDW.INC%(6),COL.INC%(
       6)
OH 370 FOR J=1 TO 6:REAO ROW.INC
       %(J),COL.INC%(J):NEXT J
NK 3B0 OATA -1,-1,0,1,1,1,1,0,0,
       -1,-1,-1
GP 390 OIM HIVE%(11,11)
EA 400 OIM USEO%(65),NOOE%(65),P
       ATH%(65),PATHLEN(20)
HK 410 SCREEN 1:CDLOR 1,2:DEFINT
        B
IN 420 OIM HEXA(100),BALL1(100),
       BALL2(100),EYES1(100),EYE
       S2(100)
NH 430 LINE (30,10)-(21,15),3:LI
       NE-STEP (0,10),3:LINE-STE
       P (9,5),3
PB 440 LINE-STEP (9,-5),3:LINE-S
       TEP (0,-10),3:LINE-STEP (
       -9,-5),3
NI 450 LINE (30,11)-(22,16),2:LI
       NE-STEP (0,9),2:LINE-STEP
        (8,4),2
JB 460 LINE-STEP (7,-4),6:LINE-S
       TEP (0,-10),6:LINE-STEP (
       -7,-4),6
FP 470 GET (21,10)-(39,30),HEXA
HE 4B0 CLS:CIRCLE (30,20),5,COLR
       (1):PAINT (30,20),COLR(1)
       :GET (25,16)-(35,24),BALL
       1
HG 490 GOSUB 5B0:GET (23,12)-(37
       ,25),EYES1
BJ 500 CLS:CIRCLE (30,20),5,COLR
       (2):PAINT (30,20),COLR(2)
       :GET (25,16)-(35,24),BALL
       2
NC 510 GOSUB 5B0:GET (23,12)-(37
       ,25),EYES2:CLS
OA 520 READ X,Y:E=(4+INT((X+7)/B
       )*Y)/2:OIM 8EE(E):8EE(0)=
       X:8EE(1)=Y:FOR I=2 TO E:R
       EAO A$:BEE(I)=VAL("&H"+A$
       ):NEXT
JL 530 OATA 26,B,B2B,A,5AA,802A,
       95AA,80AA
HK 540 OATA 952A,AA,B002,A0,1500
       ,0,0,0
NP 550 OATA 500,0,0
NL 560 RETURN
BN 570 REM PARTS
KF 580 CIRCLE (26,19),2,1:CIRCLE
        (34,19),2,1
EF 590 PAINT (26,19),1:PAINT (34
       ,19),1
IE 600 PSET (29,17):LINE-STEP (-
       2.5,-5):LINE-STEP (-2.5,3
       )
OC 610 PSET (31,17):LINE-STEP (2
       .5,-5):LINE-STEP (2.5,3)
AA 620 CIRCLE (30,24),1,1:PAINT
       (30,24),1
NG 630 RETURN
DE 640 REM GETNAMES
GB 650 LOCATE 12,16:PRINT "BeeHi
       ve":PUT (84,B4),EYES1:PUT
        (192,B6),EYES2
HC 660 FOR I=1 TO 2:LOCATE 19+I*
       2-1,6:PRINT "Player"I"'s
       name";
NI 670 INPUT PLAYER$(I):PLAYER$(
       I)=LEFT$(PLAYER$(I),15):N
       EXT I
IN 6B0 CLS:RETURN
JC 690 REM ORAWSCREEN
LO 700 CLS:Y=7
MF 710 FOR R=1 TO 11
CO 720 X=90-R*9
EC 730 FOR C=1 TO 11
IO 740 X=X+1B
BN 750 PUT (X,Y),HEXA,OR
KN 760 NEXT C
HC 770 Y=Y+15
OH 7B0 NEXT R
GL 790 PSET (297,12),2:GOSUB 930
       :LINE-STEP (0,10),2
FH B00 PSET (29B,12),2:GOSUB 930
       :LINE-STEP (0,10),2
GG B10 PSET (299,12),2:GDSUB 930
       :LINE-BTEP (0,10),2
NJ B20 PSET (96,12),2:GOSUB 930:
       LINE-STEP (0,10),2
NH B30 PSET (97,12),2:GDSUB 930:
       LINE-STEP (0,10),2
DF B40 PSET (9B,12),2:GOSUB 930:
       LINE-STEP (0,10),2
CB B50 Y1=-5:Y2=5:PSET (99,9),3:
       GOSUB 990
NL B60 PSET (99,10),3:GOSUB 990
CK B70 PSET (100,11),3:GOSU8 990
CO BB0 Y1=5:Y2=-5:PSET (9,173),3
       :GOSU8 990
AG B90 PSET (9,174),3:GOSU8 990
AE 900 PSET (9,175),3:GOSUB 990
NF 910 RETURN
QO 920 REM UPNOOWN
HC 930 FOR J=1 TO 10
HC 940 LINE-STEP (0,10),COLR(1)
CG 950 LINE-STEP (-9,5),COLR(1)
OF 960 NEXT J
NB 970 RETURN
PO 9B0 REM ACROSS
GF 990 FOR J= 1 TO 11
KA 1000 LINE-STEP (9,Y1),COLR(2)
LH 1010 LINE-STEP (9,Y2),COLR(2)
GA 1020 NEXT J
IE 1030 RETURN
PM 1040 REM SET PIECE
NB 1050 USEO=0
LF 1060 IF HIVE%(ROW,COL)<>0 THE
        N USEO=1:RETURN
LL 1070 HIVE%(ROW,COL)=PLAYER
FA 10B0 PUT (CB,RB),BEE:IF PLAYE
        R=1 THEN PUT(CB+1,R8-1),
        8ALL1 ELSE PUT(C8+1,R8-1
        ),8ALL2
JG 1090 RETURN
GJ 1100 REM CHECKLINE
BI 1110 POSSIBLE=1
CJ 1120 IF PLAYER=1 THEN 1200
EL 1130 FOR ROW=1 TO 6:FF=0:FB=0
IN 1140 FOR COL=1 TO 11
GJ 1150 IF HIVE%(ROW,COL)=PLAYER
         THEN FF=1
JC 1160 IF HIVE%(12-ROW,COL)=PLA
        YER THEN FB=1
LE 1170 NEXT COL
HM 11B0 IF FF=0 OR FB=0 THEN POS
        SIBLE=0:ROW=6
PE 1190 NEXT ROW:RETURN
CK 1200 FOR COL=1 TO 6:FF=0:FB=0
JP 1210 FOR ROW=1 TO 11
FC 1220 IF HIVE%(ROW,COL)=PLAYER
         THEN FF=1
GP 1230 IF HIVE%(ROW,12-COL)=PLA
        YER THEN F8=1
OB 1240 NEXT ROW
HO 1250 IF FF=0 OR FB=0 THEN POS
        8I8LE=0:COL=6
LD 1260 NEXT COL
JE 1270 RETURN
EB 12B0 REM CHECKWINNER
OA 1290 LOCATE 24,1:PRINT "Check
        ing...              ";
EO 1300 USEO.CNTR=0:WINNER=0:NOO
        E.CNTR=0:NOOE.TOTAL=0:CO
        UNTER=0
LD 1310 IF PLAYER=1 THEN 1440
II 1320 FOR COL=1 TO 11:ROW=1
JF 1330 IF HIVE%(ROW,COL)<>PLAYE
        R THEN 1410
NO 1340 NOOERDW=RDW:NDDECOL=COL:
        GOSUB 1560
GO 1350 IF USEO.FLAG=1 THEN 1410
EG 1360 NOOE.TOTAL=1:PATH.TOTAL=
        1:COUNTER=1
OL 1370 PATH%(1)=100*NOOEROW+NDD
        ECOL
NL 13B0 GDSUB 1650
FJ 1390 IF WINNER=1 THEN COL=11
JP 1400 REM SKIP2
LI 1410 NEXT CDL
JJ 1420 RETURN
EL 1430 REM CHECK1
FI 1440 FOR RDW=1 TO 11:COL=1
BB 1450 IF HIVE%(ROW,COL)<>PLAYE
        R THEN 1530
NB 1460 NOOEROW=ROW:NOOECOL=COL:
        GOSUB 1560
MO 1470 IF USED.FLAB=1 THEN 1530
FO 14B0 NDDE.TOTAL=1:PATH.TOTAL=
        1:COUNTER=1
OO 1490 PATH%(1)=100*NODEROW+NDO
        ECOL
KH 1500 GDSUB 1650
```

```
KB 1510 1F W1NNER=1 THEN RDW=11
JJ 1520 REM SK1P1
OE 1530 NEXT RDW
JB 1540 RETURN
CC 1550 REM USEDLDDKUP
QK 1560 USED.FLAG=Ø:SEARCH=1ØØ*N
        DDERDW+NDDECDL
ND 1570 LK=Ø:1F USED.CNTR=Ø THEN
         162Ø
FC 1580 FOR LK=1 TO USED.CNTR
KL 1590 1F SEARCH=USED%(LK) THEN
         USED.FLAG=1:LK=USED.CNT
        R
AA 1600 NEXT LK
OP 1610 REM SK1PSEARCH
OA 1620 IF USED.FLAG=Ø THEN USED
        .CNTR=USED.CNTR+1:USED%(
        USED.CNTR)=SEARCH
JA 1630 RETURN
MH 1640 REM CHECKPATH
MO 1650 NODE.CNTR=Ø
PK 1660 FOR NC=1 TD 6
AO 1670 NODEROW=NDDEROW+RDW,INC%
        (NC):NODECDL=NDDECDL+CDL
        .1NC%(NC)
PI 1680 1F NODEROW<1 OR NODEROW>
        11 DR NDDECDL<1 OR NODEC
        OL>11 THEN 175Ø
JL 1690 IF H1VE%(NODEROW,NODECDL
        )<>PLAYER THEN 175Ø
OA 1700 GOSUB 156Ø:1F USED.FLAG=
        1 THEN 175Ø
QO 1710 NODE.CNTR=NDDE.CNTR+1
OM 1720 NODE.TDTAL=NODE.TOTAL+1:
        NODE%(NODE.TDTAL)=1ØØ*ND
        DEROW+NDDECOL
JA 1730 1F (PLAYER=2 AND NODERDW
        =11) OR (PLAYER=1 AND ND
        DECOL=11) THEN W1NNER=1:
        PATH.TDTAL=PATH.TDTAL+1:
        PATH%(PATH.TDTAL)=1ØØ*NO
        DEROW+NODECOL:NC=6
KP 1740 REM SK1PNODE
MH 1750 NEXT NC
JA 1760 IF WINNER=1 THEN RETURN
QO 1770 1F NODE.CNTR=Ø AND NODE.
        TDTAL=Ø THEN RETURN
MK 1780 1F NODE.CNTR=Ø THEN PATH
        .TDTAL=PATH.TOTAL-PATHLE
        N(COUNTER):PATHLEN(COUNT
        ER)=Ø:COUNTER=COUNTER-1
AB 1790 1F NODE.CNTR>1 THEN COUN
        TER=COUNTER+NODE.CNTR-1
KP 1800 NODEROW=1NT(NODE%(NODE.T
        DTAL)/1ØØ)
EM 1810 NODECOL=NODE%(NODE.TDTAL
        )-1ØØ*NODEROW
EO 1820 PATH.TDTAL=PATH.TDTAL+1
BA 1830 PATHLEN(COUNTER)=PATHLEN
         (COUNTER)+1
BA 1840 PATH%(PATH.TDTAL)=NDDE%(
        NODE.TDTAL)
LL 1850 NODE.TOTAL=NODE.TDTAL-1
BB 1860 BOTD 165Ø
OJ 1870 REM DRAWPATH
MF 1880 LOCATE 1,1:PRINT "THE W1
        NNER: ";:PR1NT PLAYER$(P
        LAYER);
JH 1890 FOR J=1 TD PATH.TDTAL
MF 1900 ROW=1NT(PATH%(J)/1ØØ):CO
        L=PATH%(J)-1ØØ*ROW:CB=CD
        L*18+38+(6-ROW)*9:RS=ROW
        *15-1
FI 1910 IF PLAYER=1 THEN PUT(CS+
        1,RB-1),BALL1,XOR:PUT (C
        S,RS-3),EYES1,OR ELSE PU
        T(CS+1,RB-1),BALL2,XOR:P
        UT (CB,RS-3),EYES2,OR
HC 1920 NEXT J
IK 1930 REM BOAGAIN
JN 1940 LOCATE 24,1:PR1NT "Want
        to play again (Y/N)?";
NE 1950 A$=1NKEY$:1F A$<>"Y" AND
         A$<>"N" THEN 195Ø
LO 1960 IF A$="N" THEN SCREEN Ø,
        Ø,Ø:W1DTH 8Ø:END ELSE CL
        S:GOTO 5Ø
```

©

Analyze! For Amiga

David Powell

Analyze!, unlike some other spreadsheets for the Amiga, is a true Amiga software product, making full use of windows, drop-down menus, icons, color, and the Amiga mouse. You insert the *Analyze!* disk when the Amiga asks for the Workbench. When you select the disk icon, a window appears containing icons for an Empty Drawer, a Trashcan, and the *Analyze!* program itself. By using the Empty Drawer and Trashcan, you can organize a spreadsheet into directories and subdirectories, and "clean house" easily when the disk gets too full.

When you select the spreadsheet icon, *Analyze!* opens a dialog window through which you can partition off memory for your spreadsheet. The default partition is 128K. If you enter a larger value (one that's reasonable for your configuration, of course), the computer reserves that amount of memory, then displays the spreadsheet screen itself. The memory partitioning scheme lets you use most, but not quite all, of the system's free memory. On my 512K system, there were 400K bytes of memory available after *Analyze!* was loaded. However, I couldn't partition off more than about 300K.

Compression Yields Extra Room

I was curious to see how big a spreadsheet I could cram into the 128K default partition. Theoretically, at one byte per cell, a square 128K spreadsheet would have about 362 cells per side (or one could just fit a one-column spreadsheet 128K cells long.) However, *Analyze!* employs the *sparse-matrix technique* to permit much bigger spreadsheets than would otherwise be possible. Only cells holding text, data, or formulae are actually stored in memory. Empty cells, such as spaces added to improve readability, are not.

So, 128K of memory holds 128K of actual data, text, and formulae—no matter how large the spreadsheet's geography grows. For example, a one-column, 128K spreadsheet could actually be 256K cells long if data cells alternated with empty cells. This permits you to arrange the spreadsheet in an attractive manner without worrying about wasted memory.

Intuitive Operation

From within the spreadsheet screen, you reveal *Analyze!*'s main menu bar by holding down the right mouse button. The menu bar contains five menus: Project, Range, Worksheet, Print, and Recalculate. While holding the right button down, move the mouse pointer to one of these options; a menu of its commands drops into view. You select a command by sliding the mouse cursor to it and releasing the mouse button. In short, *Analyze!* handles menus and other program options in the usual Amiga fashion, which will seem natural to Amiga owners. It's easy to take these intuitive, easy-to-use features for granted until you try operating an Amiga program that lacks them. (It's still possible to buy an Amiga program that doesn't look or act like Amiga software at all. Amiga programs that ignore the mouse and visual icons, operating chiefly through keyboard controls, are usually quick translations of software written for an older machine such as the IBM PC.)

Commands within the Project menu display a Worksheet's current formatting parameters and allow you to load, store, delete, and update spreadsheets stored on internal or external disk drives. (Spreadsheets can be stored on disks used by other programs, because *Analyze!* only looks for files with the extension .SHT.)

The Range menu offers commands that name, format, label, copy, move, erase, and write-protect individual cells or groups of cells. People building spreadsheets will use these functions frequently, and it's nice to have them all in one place.

Moving And Copying Cells

An example will show you how easy the Range command—and Amiga's mouse—make the task of moving or copying a block of cells to a new location. This requires only three steps:

1. Select the Range option's Move (or Copy) command. A prompt appears on the screen asking for the range of cells you want to move.
2. Position the mouse cursor at the upper-left cell of this range, press the left mouse button, and drag the cursor to the lower-right cell. Release the mouse button; a prompt appears on the screen asking for the move destination.
3. Move the mouse pointer to the upper-left cell of this destination; then click the left button. *Analyze!* repositions the entire block of cells so that its upper-left corner coincides with the destination cell.

I like the fact that such operations can be done without touching the keyboard. However, you can't use the mouse to define ranges that go beyond the visible screen. So *Analyze!* also offers simple keyboard procedures for selecting ranges and jumping to different places in a spreadsheet.

When you copy cells to a new location, *Analyze!* can copy formulae in the cells in *absolute* form (with row and column references transferred verbatim), in *relative* form (with references adjusted for the new location), or in a combination of both. (However, all cell references are kept verbatim when you transfer formulae to a new location with a Move command.)

A Variety Of Formats

The main menu's Worksheet option includes commands that insert or delete blank rows and columns, erase a spreadsheet, enter titles, format all cells, set column widths, justify labels, and write-protect the entire spreadsheet. Of special interest is the Worksheet option's Format command, which differs from the Range option's Format command in scope. Worksheet formatting applies to every cell in the entire sheet, not to a specific block of cells.

Through Range-Format and Worksheet-Format, you can display data in the following formats:

- fixed-point decimal
- scientific (exponential) notation
- dollars and cents

- percentages
- dates
- with commas (for instance, 2,123 instead of 2123)

Negative numbers are automatically displayed in red to distinguish them from positive numbers, which appear black on the paper-white background of the spreadsheet.

Another Worksheet-Format option (labeled as +/−) can convert positive and negative integers into crude bar charts. This option is designed to work only with integers (whole numbers), so it doesn't work as well with noninteger values.

Following Worksheet in the main menu is the Print command, which enables you to format a spreadsheet and send it to a printer. (However, you must still use Preferences to select the correct settings for your particular printer.) The Print feature allows you to set top-of-form, define page lengths, transmit linefeeds, print part or all of a spreadsheet, set all four page margins, define page headers and footers, pick rows or columns to use as page borders, and print calculated formula results or the formulae themselves. If you don't want to print directly to a printer, you may send the same output to an ASCII disk file for further formatting by a word processing program.

The last option in the main menu (Recalculate) lets you set your spreadsheet's calculation order. You can make recalculations automatic (after each cell change) or manual (as requested). The calculation order can be top-to-bottom or right-to-left. Or, it can be natural, in which case the system performs multiple passes to pull together complex data relationships the way a person would.

This offers more calculation flexibility than many spreadsheets I've seen, but there's even more. *Analyze!* also lets you create a spreadsheet that runs through as many as 50 iterations, or recalculations, before displaying its results. As a former mathematician, I value this feature highly.

Special Functions

Advanced users will also welcome the program's library of special functions. These include, but are not limited to, the following:

- comparisons and logical operators
- trigonometric functions
- statistical averages, standard deviations, and variances
- table lookups within a spreadsheet
- logarithms and exponentials
- present/future values of cash flows
- loan and annuity payments
- maxima/minima of values in a block
- modulus arithmetic
- random numbers

Analyze! is an effective, efficient spreadsheet, with very few apparent bugs. However, I do have some small complaints. It does not, for instance, offer a macro capability for writing spreadsheet-template programs (power users, take note). It could also handle formula input better. Some spreadsheets use a parser that looks at what you type and decides on its own whether you have entered data, text, or a formula. *Analyze!*, on the other hand, makes you begin every formula with a plus sign (+). This is a bit awkward.

The *Analyze!* user manual, like others of its type, suffers from too much text and too few illustrations. You should follow along with the computer as you read the manual. However, it does include very useful summaries of all system menus and special functions. However, since *Analyze!* is so well integrated with the Amiga's Workbench metaphor, you can learn to use the program almost without opening the manual.

Analyze!
Micro Systems Software
4301-18 Oak Circle
Boca Raton, FL 33431
$99.95

The American Challenge: A Sailing Simulation

Tony Roberts

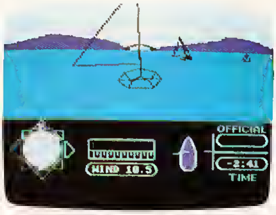
Requirements: Apple II-series computer with at least 64K RAM. IBM PC or PCjr with 128K RAM and DOS 2.0 or above. Graphics card required for use with PC. Commodore 64 (available early fall).

The pleasures of sailboat racing are effectively recreated in *The American Challenge: A Sailing Simulation* from Mindscape and Tom Snyder Productions. Fashioned after the America's Cup races, the goal of the game is to win all the preliminary heats. This, in turn, gains you the right to challenge the Australians in an attempt to regain the Cup for the United States. Should you manage to beat Australia in the program's Cup Race, you become eligible for a contest that could win you a trip to Australia to watch the 1987 America's Cup races in person (the contest closes on October 30, 1986).

Taking The Challenge

To play the game, you choose a course; the computer displays an overhead view of the course and shows you a suggested route around it. Sailing against a boat piloted by the computer, you jockey for position and attempt to cross the starting line just as the horn sounds.

The computer sails a pretty good race. It's possible, but not easy, to beat it, and there's little room for error if you hope to win. You control your boat's direction, sail trim, and centerboard position. At any time during the race, you can press the space bar to return to the overhead view, which shows the paths both boats have taken. Press the space bar again and the race resumes. Other controls allow you to look right and left off your board and to zoom in on the competition or zoom back for a wider angle view.



The American Challenge: A Sailing Simulation *recreates the challenge of competing in the America's Cup races.*

Seven of the eight courses are based on the courses used in actual sailboat races. Each race becomes progressively more difficult as the currents become stronger and your compass is taken away.

You're not to sail the Cup Race until your boat has beaten the computer at all seven of the preliminary races. Even for someone familiar with sailboat racing, it will take quite a while to become that proficient.

Racing against the computer is a challenge, but also becomes predictable. The computer maintains a record of the best time for each course and sails a course the same way each time until it is beaten.

Two-Computer Version

One way to eliminate this predictability is to choose the two-player option. However, this choice requires that you have two computers connected by modem or a null modem cable, and both computers must be running the program. With this option, you can send messages to the other captain. This communication becomes necessary to settle disputes regarding collisions or possible rules violations.

Sailing against another human adds to the enjoyment of the game, but it also slows things down a bit. If you are using 300 bit-per-second modems, the races take from five to twenty minutes each.

One other option allows you to race a high-speed motorboat around the courses. This can be fun, but don't expect to take on the Australians with anything but wind power.

While explaining the program, the manual also imparts quite a bit of information about sailing itself, including sailing basics, racing strategy, and right-of-way rules. The package even includes a 45 r.p.m. phono record with a sailing tutorial for novices.

The American Challenge: A Sailing Simulation
Mindscape
3444 Dundee Road
Northbrook, IL 60062
Apple II series/IBM flippy version
$39.95
Commodore version (available early fall)
$29.95

Vorpal Utility Kit

N. Randall

Requirements: Commodore 1541 disk drive.

It has never been any secret that a major problem with a Commodore 64 system is the speed of the disk drive. It's slow. Several companies, understanding the impatience of the regular 1541 user, have released products that speed it up. One of the most popular has been Epyx's *Fast Load* cartridge, which many owners now swear they could scarcely do without. Following the success of *Fast Load*, Epyx has now released the *Vorpal Utility Kit*. For anyone who needs to manipulate files, copy disks, or make use of extremely fast loads and saves, the *Vorpal* package could quickly become indispensable.

The *Vorpal Utility Kit* is actually several utilities in one. With *VFiler*, you can load and save user-created programs at about 25 times the normal 1541 speed. Note that this does not apply to commercial software; the *Fast Load* cartridge takes care of those. What the *Vorpal* kit does is add a fifth file type to the 64's normal four (program, sequential, user, and relative). These files make use of the kit's greatly increased speed.

As a nonprogrammer, I must confess to a thorough disinterest in these super-fast files, simply because I never create programs that could use them. They can be used, though, with any BASIC program (and some ML programs) which you receive from user groups or type in from a book or magazine, in addition to those you create yourself. Epyx makes it clear on the package that the high speed applies only to user-created software and BASIC programs.

20-Second Formatting

More exciting, for nonprogrammers at least, are the disk and file utilities. With the *Vorpal Utility Kit*, you can format a disk in 20 seconds rather than the usual two minutes. And you can copy an entire disk—including formatting—in less than three minutes. For those with two or more disk drives, the software allows you to renumber both the origin and the destination drives as needed.

File commands include Delete, Undelete, Protect, Unprotect, and Rename, in addition to the following special functions. You can change a file from one type to another. For example, if your old word processor stores documents as USR files, and you buy a new word processor that stores them as PRG files, the *Vorpal Utility Kit* lets you change them in seconds, without the tedium of reading a file into memory and writing it back to disk in the new format. You can also copy files and convert them at the same time.

The final utility in the *Vorpal Utility Kit* is a hardware check. The program will check your 1541's head alignment and drive speed, and will even attempt to correct a minor alignment problem. Impressively, all the commands on the *Vorpal* kit respond to the touch of a single key, and the manual, although certainly complete, is practically superfluous. Even if you use only the 20-second formatting or the three-minute disk copying, the *Vorpal Utility Kit* is one package you will not want to pass up.

Vorpal Utility Kit
Epyx
1043 Kiel Court
Sunnyvale, CA 94089
$34.95

Lords Of Conquest

Todd Heimarck

Lords of Conquest from Electronic Arts is a lot like the popular board game Risk, and in some ways, it's even better.

All the exciting, entertaining, and educational games, applications, and utilities from COMPUTE! magazine are now available on disk

for your Commodore, Atari, Apple, or IBM personal computer.

The COMPUTE! Disk

A new *COMPUTE! Disk* is published every month, rotating among the four major machines covered by *COMPUTE!*: Commodore 64 and 128; Atari 400/800, XL, and XE; Apple II-series; and IBM PC, PCjr, and compatibles.

Every three months you can receive a disk with all the quality programs from the previous three issues of *COMPUTE!* that will run on your brand of computer.

Like the popular *COMPUTE!'s Gazette Disk*, the *COMPUTE! Disk* is ready-to-load and error-free. It saves you valuable hours of typing time and eliminates typing errors.

With a subscription, you will receive one disk every three months for a total of four disks a year—for only $39.95. That saves you $20 a year off the single-issue cost.

Or you can order individual issues of the *Disk* for $12.95 a disk plus $2.00 shipping and handling.

Remember to specify your type of computer when ordering the *COMPUTE! Disk*. You'll find more information about this month's *COMPUTE! Disk* in this issue. (Note: You'll need the corresponding issues of *COMPUTE!* magazine to use the *Disk* since the disk will have no documentation.)

For fastest service when ordering a subscription to the *COMPUTE! Disk*, call toll free 1-800-247-5470 (in Iowa 1-800-532-1272).

For more details or to order individual issues of the *COMPUTE! Disk*, call our Customer Service Department toll free at 1-800-346-6767 (in New York 212-887-8525).

Please allow 4–6 weeks after placing an order for your first disk to arrive.

COMPUTE! Publications,Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York NY 10019
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk, COMPUTE! Books, COMPUTE!'s Apple Applications, and COMPUTE!'s Atari ST Disk and Magazine

King Of The World

A game of Risk begins with a world map divided into a number of countries owned by various players. By shaking the dice, you win and lose territories. Some countries are isolated (Eastern Australia, Japan, and Argentina), while others are busy crossroads (the Middle East and the Ukraine, to name a couple). The ultimate goal is to build up your armies and win enough battles to conquer the world.

In *Lords of Conquest*, the basic idea is to take over the world, but you win by building or capturing a certain number of cities—from three to six. Some of your territories produce raw materials such as gold, iron, coal, timber, and horses. When you've acquired certain combinations of materials, you can buy weapons or place a new city on the map.

Before the game starts, you split up the available territories. It's important to choose countries that contain coal mines, gold mines, forests, and the like, so you can start building up your stockpile of raw materials. At the same time, you should pick areas that are near each other, because your defenses will be stronger if you have friendly countries as neighbors.

Up To Four Players

You can play one-on-one against the computer, or you can involve as many as four human players. The disk contains 20 maps, including Europe, Africa, North America, the Middle East, South America, Japan, Australia, and the Mediterranean. If you're not satisfied with the built-in maps, you can ask the computer to generate a random battlefield from parameters you supply. You can also create your own map. It takes some time to build a map, but you can fine-tune it until it looks just the way you want. These new maps can be saved to disk for use in later games.

Select a level of play: beginner, intermediate, advanced, or expert. In the beginner level, there are only pastures (a source of horses) and gold mines; this level is suitable for playing with children. More challenging is the expert level, featuring horses, gold, timber, coal, and iron.

Should you choose to play the computer, you must also select a level of difficulty. Level 1 gives you a big advantage (four extra territories) and level 9 skews the game in favor of the computer.

After you divvy up the territory, the game begins. Each round has several phases. During development, you can use your gold and other commodities to create weapons, boats, or cities. Production comes next; more raw materials are added to your inventory. You then have a chance to move your stockpile to a new country. The stockpile is like an imperial treasury; if another player captures it, he or she will get all your gold, iron, coal, and timber. Finally, there's a combat phase during which each player can send forces against the other players. You're limited to two attacks per round.

To create a city, you have two choices: Spend one unit of iron, coal, timber, and gold, or use four gold units. In the advanced and expert games you can build a boat (a naval force) with three timber units, or buy one with three units of gold. A boat can carry a horse and a weapon, which makes it a valuable offensive force.

Offense Or Defense?

There's a lot to be said for building cities. The ultimate goal is to own three or more cities, so each one you build brings you one step closer to winning. Cities also increase production in the neighboring countries. If you place a city next to a gold mine, its output will double from one unit to two.

But cities are fairly expensive. And if you spend all your resources on cities while your opponents build up their horses, weapons, and boats, you may eventually lose the game. Your opponent will likely attack and conquer your cities. Ownership of a certain number of cities is the goal. It doesn't matter whether you build the cities or capture them.

Each game of *Lords of Conquest* has a definite rhythm. In the first couple of rounds, weak and isolated countries are overrun by invaders, especially if the country produces a valuable commodity. As the territories coalesce in the middle rounds, powerful armies build up along the borders between empires. When boats first appear, the complexion of the game changes. Suddenly, any coastal country is vulnerable to an attack from the sea. It's difficult to defend a coastal country from marauding Vikings.

The mechanics of the game are fairly simple; there are four commodities, three weapons, and the cities. But *Lords of Conquest* requires a good sense of strategy. On your way to the goal of building cities, you have to watch your resources and try to keep them from your opponents. If you own no country with a gold mine, you may have to develop a short-term strategy to capture one. You should spend your money wisely, occasionally forgoing a new weapon to save up for a city.

Geography and distribution of resources are also important factors. The strategy that works best on one map might fail miserably on another. Boats are valuable when islands are plentiful, but they're relatively unimportant when the map contains mostly land.

The Role Of Diplomacy

The computer plays a tough game; at the higher levels you won't often beat it. And when you play with other people, diplomacy plays a role: "I won't attack you if you won't attack me." The multiplayer game also allows for alliances. When more than two players are near a battle, the uninvolved players can send forces to the attacker or defender, or they can remain neutral. You also have a chance to trade commodities—a gold and an iron for two coal mines, for example.

If you're a Risk player, you'll enjoy *Lords of Conquest*, and if you get tired of conquering one world, you can easily find or build another. A second useful feature is the one-player game: When you want to play, but can't round up a group of opponents, you can test the computer's abilities. The only negative comments I've heard concern the graphics. There's nothing particularly wrong with them; they're just simple. The countries, for example, are made up of colored squares. This doesn't affect the playability of the game, so it's a minor criticism.

Lords of Conquest
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
Commodore version $32.95
Apple II and Atari 8-bit versions soon to be released; no prices available.

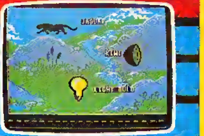
©

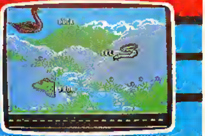
ILS INTERACTIVE LEARNING SERIES

FIREBIRD

TALKING TEACHER

SOFTWARE THAT TALKS AGES 2–8





Talking Teacher combines revolutionary software speech, superb graphics and engrossing play action to provide you with the next step in home computer software. No special hardware is required to make your computer talk! Talking Teacher contains three seperate learning games for children ages 2–8. The lessons teach:

- Capital and lower case letter identification and keyboard location
- A basic vocabulary and initial letter sounds of over 100 words.
- Spatial relationships (left, right, higher and lower).

Lesson One pronounces and prints each letter as it is typed. Children can type their names, short messages or just have fun with the keyboard! Ages 2–5.

Lesson Two Leads your child through the alphabet, pronouncing each letter and using it in a word. Typing the right letter causes a countryside filled with fantastic creatures and objects to unfold. Ages 3–6.

Lesson Three asks your child to identify the beginning letter of the words presented in Lesson Two. Mistakes are reviewed and correct answers are musically rewarded. Ages 4–8.

NOTE TO PARENTS: After loading, all three lessons may be enjoyed by your children without your direct supervision. However, the educational value can be enhanced by your participation and encouragement.

FACTSHEET/ORDER HOTLINE: 201.934.7373 FIREBIRD LICENSEES INC. P.O. BOX 49, RAMSEY, N.J. 07446

COMMODORE 64/128 ARE TRADE MARKS OF COMMODORE BUSINESS MACHINES.

COMMODORE 64/128K

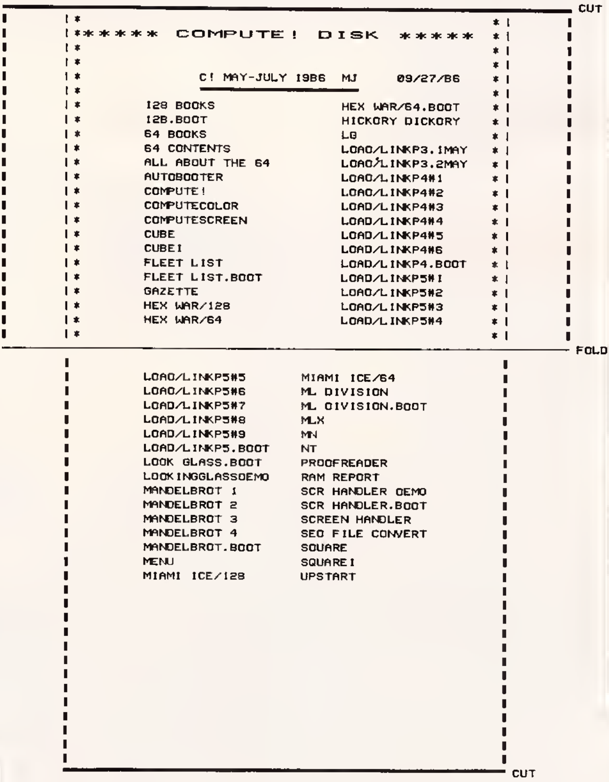
Jacket Lister

Gregory Jackmond

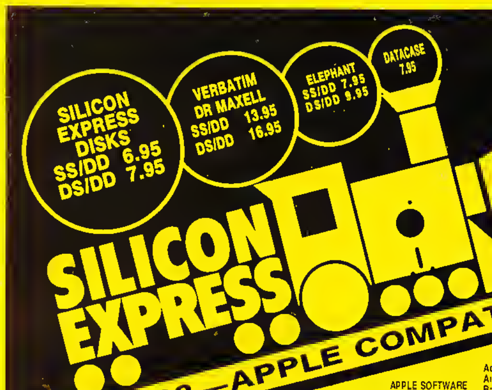
The more disks you have, the more you'll enjoy this novel utility. It prints a disk jacket with an alphabetized directory of all the programs on a disk. The original version of "Jacket Lister" runs on the Commodore 64 and 128 (in 64 mode). We've added new versions for the IBM PC/PCjr, Apple II series (DOS 3.3 and ProDOS), and Atari 400, 800, XL, and XE. A printer is required. The Atari version requires at least 32K of memory.

How many times have you picked up a disk, only to realize that you can't remember which programs are on it? You can always get a disk directory in the usual way—by putting the disk in the drive and listing the directory on the screen—but that's slow and tedious when you're looking for a specific program.

"Jacket Lister" is a unique, time-saving solution to this perennial problem: It not only allows you to create a personalized jacket out of ordinary paper, but also lists an alphabetized directory on the jacket itself. In a glance, you can see which programs are on each of your disks. A date is also included so that you can tell whether the listing is obsolete. The jacket listing may include as many as 88 filenames, using the front and back of the jacket. (Some computers can store more than 88 files on a disk, but the jacket does not have room for more than that number.)



```
                                                                    CUT
|  |*                                                        *|  |
|  |****** COMPUTE! DISK ******                              *|  |
|  |*                                                        *|  |
|  |*                                                        *|  |
|  |*          C! MAY-JULY 1986  MJ      09/27/86            *|  |
|  |*                                                        *|  |
|  |*      128 BOOKS                 HEX WAR/64.BOOT         *|  |
|  |*      128.BOOT                  HICKORY DICKORY         *|  |
|  |*      64 BOOKS                  LG                      *|  |
|  |*      64 CONTENTS               LOAO/LINKP3.1MAY        *|  |
|  |*      ALL ABOUT THE 64          LOAO/LINKP3.2MAY        *|  |
|  |*      AUTOBOOTER                LOAO/LINKP4#1           *|  |
|  |*      COMPUTE!                  LOAO/LINKP4#2           *|  |
|  |*      COMPUTECOLOR              LOAD/LINKP4#3           *|  |
|  |*      COMPUTESCREEN             LOAD/LINKP4#4           *|  |
|  |*      CUBE                      LOAD/LINKP4#5           *|  |
|  |*      CUBEI                     LOAD/LINKP4#6           *|  |
|  |*      FLEET LIST                LOAD/LINKP4.BOOT        *|  |
|  |*      FLEET LIST.BOOT           LOAD/LINKP5#I           *|  |
|  |*      GAZETTE                   LOAO/LINKP5#2           *|  |
|  |*      HEX WAR/128               LOAO/LINKP5#3           *|  |
|  |*      HEX WAR/64                LOAD/LINKP5#4           *|  |
|  |*                                                        *|  |
---------------------------------------------------------------- FOLD
   |                                                     |
   |       LOAO/LINKP5#5       MIAMI ICE/64              |
   |       LOAO/LINKP5#6       ML DIVISION               |
   |       LOAD/LINKP5#7       ML OIVISION.BOOT          |
   |       LOAD/LINKP5#8       MLX                       |
   |       LOAD/LINKP5#9       MN                        |
   |       LOAD/LINKP5.BOOT    NT                        |
   |       LOOK GLASS.BOOT     PROOFREADER               |
   |       LOOKINGGLASSOEMO    RAM REPORT                |
   |       MANDELBROT 1        SCR HANDLER OEMO          |
   |       MANDELBROT 2        SCR HANDLER.BOOT          |
   |       MANDELBROT 3        SCREEN HANDLER            |
   |       MANDELBROT 4        SEO FILE CONVERT          |
   |       MANDELBROT.BOOT     SOUARE                    |
   |       MENU                SQUAREI                   |
   |       MIAMI ICE/128       UPSTART                   |
   |                                                     |
   ------------------------------------------------------- CUT
```



INSIDE OHIO
1-800-225-0755

YOUR TOLL FREE ACCESS OUTSIDE OHIO

1-800-228-0755

CALL...FOR THE BEST PRICES...
BEST DELIVERY...
We have hundreds of other
programs in stock at
Fantastic savin s.

LASER 128—APPLE COMPATIBLE COMPUTER $395.95

	IBM
Ability	299 95
ASC Il Expiess	132.95
Bank Street Writer	46.50
Clout	119.95
Copy Il Option Bd.	74.00
Cornerstone	259.95
Crossstalk	94 95
dBase III -	419.95
Deadline Manager	104 95
Dollars & Sense	104.95
Easy	97.50
Einstein Writer	118.95
Electric Disk	214.95
Fontrix 2.5	84.95
Fntpks 1 thru 13	ea. 13.00
Framework Il	419.95
Gamecard III	39 95
Gato	23.69
In House Accountant	119.95
Jef	32.75
Knowledgeman 2.0	319.95
Lotus	CALL
Mngng Yr Money	104.95
Micro Cookbook	22.00
Microsott Chart 2.0	299.95
Woid	239.95
Windows	69.95
Norton Utilities	55.95
Option Boaid	74.00
PFS: Access	59.95
PFS: Files	82.50
PFS: First Success	229.95
PFS: Graph	82.50
PFS: Plan	82.50
PFS: Pioof	59.95
PFS: Report	82.50
PFS: Wiite	82.50
Print Shop	36.95
Print Shop Gr. #1	22.95
Print Shop Retill	12 25
R:Base 5000	339.95
Retlex	89.95
Remote	92.00
Report Card	32.95
Samma +	369.95
Sidekick protected	29.00
Sidekick not protected	44.00
Sideways Vei. #3	42.00
Speedkey	104.95
SupeiCalc 3	239.95
Superkey	35.50
Symphony	CALL
Think Tank	99.95
Time Is Money	69.95
Traveling Sidekick	36.95
Turbo Database Tlbx	38.95
Tuibo Lighting	54.50
Turbo Pascal 3.0	36.25
Tuibo Pascal 8087	55 00
Tuibo Prolog	69.95
Turbo Tutor	19 00
Ultima III	34 95
Visable 8088	35 95
Volkswiiter Dlx #3	141.00
Web. Spell Checker	41.95
Will Writer	27.95
Wizardry	36.00
Word Pertect	219.95
Wordstai	214.95

	APPLE SOFTWARE
Adv. Construction	30.50
Algeblastei	27.00
Alpha Plot	24.89
ASC II Express	72.50
Auto Works	27.95
Bag ot Tricks 2	28.00
Bank Street Filer	38.95
Bank Street Mailei	38.95
Bank Stieet Writer	38.95
Bank Street Speller	38.95
Baid's Tale	27.50
Beagle Bag	14.95
Beagle Basic	17.95
Beagle Graphics	29.95
Beneath Apple DOS	13.95
Big U	16.95
Bookends	64.95
Captain Goodnight	19.95
Carmen Sandiego	22.45
Chmpshp Ld. Runner	19.45
Chart n Graph Tlbx	23.95
Computer Quarterback	27.95
Complete Giaphics	27.95
Crypt ot Medea	21.95
Database Toolbox	23.95
Dazzle Draw	33.95
D-Code	19.95
DiskQuick	14.95
Dollars & Sense	66.50
DOS Boss	11.95
Double-Take	17.95
Eidolon	23.50
Essential Data Dup	41.95
Extra K	19 95
Fantavision	26.95
Fat Cat	17.95
Flex Type	14.95
Fontrix 1.5	53.25
Fntpks 1 thru 13	ea. 13.00
Font Works	34.95
Forecast	35.00
Frame-Up	14.95
Gato	20.45
G I Joe	19.95
Graphics Department	64.95
Graphics Expander	21.95
Graphic Magician	26.95
Graphworks	57.95
GPLE	24.95
Handlei Package	69.95
Impossible Mission	23.50
I.O. Silver	14.95
Jeeves	29.95
Karateka	19.45
Knight of Diamonds	19.50
Kids on Keys	17.95
Kid Writei	18.89
King's Quest I or II	28.009.95
Koronis Ritt	23.49
Legacy of Llylgamyn	22.50
List Handlei	39.95
Locksmith	56.95
Mngng Yi Money	101.95
Megawoiks	64.95
Meilin	33.69
Merlin Pro	54.95
Merlin Combo	63.95
Micro Cookbook	21.45
Millionaire	22.50
Minipix	14.69
Moebius	34.95
MouseWrite	71.95

	APPLE SOFTWARE
Munch-A-Bug	23.95
NATO Commandei	19.45
PFS: Plan	69.95
PFS: Access	44.00
PFS: Write	69.95
PFS: File	69.95
PFS: Report	69.95
PFS: Giaph	69.95
PFS: Fiist Success	126 00
Piece ot Cake Math	15 95
Pinpoint	39.95
Prince	48.95
Printographer	23.95
Print Shop	26.69
Pit Shp Gr. #1.2. or 3	13.95
Print Shop Comp.	21.95
Probytei	17.95
Pionto DOS	14.95
Prot tour golf	25.89
Quicken	51.95
Report Card	32 95
Rescue/Fiactalus	23.50
Reportworks	64 95
Rescue Raideis	24 95
Science tool kit	32.95
Sensible Grammar	52.95
Sensible Spellei	64.95
Shape Mechanic	20.00
Sideways	29.95
Silicon Salad	12.69
Skyfox	24.25
Spellworks	29.95
Spitfire Simulatoi	24.75
Spy Huntei	26.00
Sticky Bear ABC	23.75
Basket Bounce	23.75
Cai Builder	23.75
Math	23.75
Numbers	23.75
Opposites	23.75
Reading	23.75
Spellgrabber	23.75
Townbuilder	23.75
Typing	23.75
Summer Games I or II	22.45
SuperCalc 3A	109 95
Terrapin Logo	54.50
Terrapin Utilities	14.95
Think Tank	88.89
Thinkworks	64.95
Time is Money	57.45
Tip Disk #t	12.00
Transylvania	11.95
Triple-Dump	18.95
Tuibo Database Tlbx	27.95
Tuibo Pascal 3.0	34.95
Turbo Tutor	20.50
Ultima III	33.25
Ultima IV	33.25
Understanding IIE	17.95
Universal tile conv	24.95
Utility City	15.25
Versaform	43.45
Video Toolbox	23.95
Visable 6502	31.75
Wilderness	38.95
Winter Games	21.45
Wizardry	26.50
Wizard's Toolbox	23.95
Wizprint	14.25
Word Handler	39.95
Word Perfect	125.95
World's Grtst Ftball	22.45
Wiite Choice	26.45

	COMMODORE
Acrojet	24 95
Aii Rescue I	24 95
Bard's Tale	27.50
Ballblazer	23.49
Back to Basics	134 95
Bank Street Storybook	27.95
Bank Stieet Speller	34.95
Bank Street Wiitei	34.95
Basic Toolkit	29.95
Blazing Paddles	24.95
Biiwed Time 64/128	20 95
Color Me	20.95
Cntdwn to Shtdwn	20.95
Fast Tracks	20.95
Gertrude's Secrets	20.95
G I Joe	19.95
Graphics Magician	27 95
Giappler CD	79.95
Grt Amei Road Race	20 95
Hackei	20.95
Jet 64/128	27.95
Jet Combat Simulator	20.95
Karateka	19.45
Kennedy Approach	24.95
Kola Pad	54.95
Kung Fu	20.95
Little Comp. People	24.95
Mail Order Monsters	24.95
Merlin	34.95
Mig Alley Ace	24.95
Multiplan 64/128	43.95
Music Shop	31.95
Nato Commander	19.75
Paper Clip	41.95
PFS: File	34.95
Print Shop	29.95
Print Shop Comp.	21.95
Pr. Shop Gr. 1,2.or 3	13.95
Rescue on Fractalus	23.50
Sciabble 64/128	27.95
Sideways	20.95
Silent Service	19.95
Spitfire Ace	20.95
Spreadsheet	34.95
Spy vs. Spy	20 95
Sticky Bear ABC	20 95
Numbers	20.95
Opposites	20.95
Shapes	20.95
Summer Games I oi II	22 45
Temple of Apshai Trit.	22.45
Winter Games	21 45
World's Grtst Bsball	19.45
World's Grtst Ftball	22 45

	IBM APPLE COMMODORE
Clip Art (Vol 1)	15.95
Clip Art (Vol 2)	21.95
Copy II -	18.50
Crossword Magic	30.00
Crusade in Europe	22.95
Cut & Paste	23.00
Cutthroats	23.69
Deadline	28.00
Early Games	15.95
Easy as ABC's	20.95
Enchanter	20.95
Family Roote	149.95
Flight Sim II	30.50
Fraction Factory	15.95
F-15 Stiike Eagle	20.00
Ghost Busters	22.00
Hitchhiker's Guide	21.75
Intidel	26.00
Load Runner	19 45
Magic Spells	20.75
Mastertype	23.25
Math Blaster	26.95
Micro Lg. Baseball	22.50
Micro Lg. Manager	27.95
Micro Lg. Team	13 95
Mind Forever Voy.	30.95
Music Construction	24 75
Newsroom	30 50
One-On-One	24.00
Pitstop II	23.50
Planettall	22.00
Print Shop Retilt	12.25
Reader Rabbit	22.75
Sargon III	27.20
Sat (Haicort Brace)	45.00
Seastalker	22.00
Soiceiei	26.00
Solo Flight	19.95
Speed Reader II	36.95
Spell It	26.95
Spy Hunter	26 00
Starcross	28 00
Suspect	26.00
Suspended	28.00
Temple ot Apshai Tri	22.45
Typing Tutor III	28.50
Will Writei	28.45
Wishbringer	27.95
Witness	22.00
Word Attack	26.95
Word Spinner	20.50
World's Grtst Bsball	19.95
Zork I	21.75
Zork II or III	24.75

	HARDWARE
Compuserve Starter	21.95
Datacase	7.95
Disk Notcher	4 95
Flip-n-File	10.95
Lemon	31.95
Lime	48.95

	APPLE HARDWARE
80 Col. 64K Caid	49 95
Apple Cat II	184.95
Disk Drive Hi Tech	139.95
Gibson Light Pen	148.00
Giappler Buttered	114.95
Grappler -	71.95
Grappler (serial)	71.95
Grappler C	71.95
Hotlink	44.95
Kratt Joystick	25.95
Koala Pad -	70.45
Laser 128 Computer	395.95
Mach II Joystick	26.20
Mach III Joystick	32.20
Micromodem IIE	139 95
Mockingboaid A or B	65.95
Mockingboard C	116.00
Mockingboard D	126.45
Mockingboaid A-MAX	38.95
Multiram CX 512K	209.00
Multiram IIE 80/64K	124.00
Multiview II - 80/160	199.00
Paddlesticks	26.50
Parallel Printer Caid	44 95
Piometheus 1200A	254.95
Print-it	123.50
Super Serial Caid	84 95
System Saver Fan	59.95
Thunder Clock	102.95
Wildcard It	74.95
ZEE 80A Checkmate	54.95
Zoom IIE	109.00

PRINTERS	
Star NX-10	CALL
Star Powertype	344.95
Printer Stand	16.95
Okidata 182P	229.95
Okidata 192P	399.95
Okimate 20	209.95
Colored prntr ribbons	CALL
Color Paper Pack	12 95

MONITORS	
Magnavox Monitors	CALL
Amdek Color 600	414.99
Samsung Color Mntr	169.96

YOUR ORDER FORM
SILICON EXPRESS
5955 E. Main St. Columbus, Ohio 43213

NAME ____________________

ADDRESS ____________________

CITY ____________ STATE ______ ZIP ______

CHARGE CARD # ____________ Exp. Date ______

QTY.	DESCRIPTION	PRICE

Computer Type ________ SHIPPING

Phone No. ____________ TOTAL

No additional suichaige on Mastercard. Visa, or School puichase oiders Personal checks allow 2 weeks. Shipping charges are $3 00 minimum Ohio iesidents add 5½% state sales tax Prices subject to change without notice

SILICON EXPRESS

5955 E. Main St. Columbus, Ohio 43213
1-614-868-6868

Type in the appropriate program for your computer, then save a copy before you run it. The variable NS$ in line 420 (NAME$ in line 15 for the Atari version) defines your personalized title for the jacket, which you can change to whatever you like. You may substitute any characters in the definition of NS$, but don't make the string longer than 26 characters.

Jacket Lister is a self-prompting program, so you don't need elaborate instructions. Simply run the program, insert the disk that you want to catalog, then follow the screen prompts to create a custom jacket for that disk. When the jacket has finished printing, all that's left to do is to cut the cover to size, fold it along the printed fold lines, and glue the flaps.

Commodore 64 Version

Commodore Jacket Lister (Program 1) runs on a Commodore 64 or Commodore 128 in 64 mode. The program is written for standard Commodore printers (and for non-Commodore printers that can emulate the standard Commodore graphics characters), but can easily be modified to work on other printers as well. Simply change the graphics symbols to dashes (–) or exclamation points (!) in lines 510, 1100, and 1240. (Horizontal lines are formed from the dashes, and vertical lines from the exclamation points.) The program also uses characters 17 and 145 as control codes to set the printer for lowercase/uppercase or uppercase/graphics printing, respectively. You may need to substitute other control codes for these in lines 100 and 110.

If you have a Commodore Plus/4, 16, PET/CBM, or VIC-20 with expansion memory, you should be able to make Jacket Lister work with only slight modifications. The POKEs that change the screen color and create sound effects are specific to the Commodore 64; if you delete these statements, the program should run on nearly any Commodore computer.

Atari Version

The Atari version (Program 2) runs on any Atari 400, 800, XL, or XE computer with at least 32K of memory, and should work with any standard-width printer. No special instructions are required; simply follow the directions on the screen.

Apple II Version

With the Apple II version of Jacket Lister, all output is in uppercase. If you are using DOS 3.3, type in Program 3 as listed. For ProDOS, start with Program 3, but omit lines 80–200 and add the lines listed as Program 4. In either case, you may have to modify line 450 to suit your particular printer configuration.

IBM PC/PCjr Version

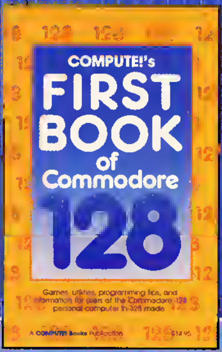
In this version of Jacket Lister (Program 5), all output is in uppercase.

Program 1: Commodore Jacket Lister

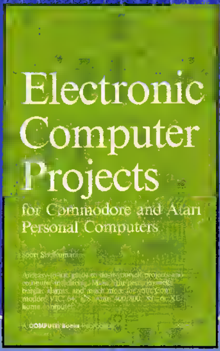
For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
GD 1Ø REM ***** PROGRAM SET UP
       ******
RH 2Ø DIMTB$(144):DIMAB$(144)
KR 3Ø PRINT"{CLR}{7}":POKE5328
      Ø,14:POKE53281,6
PS 4Ø PRINT"{4 DOWN}{9 RIGHT}
      {RVS}{WHT} WHAT IS TODAY
      'S DATE: ":PRINT:
EK 5Ø PRINT"{5 RIGHT}{RVS}
      {WHT} ENTER MO/DY/YR THE
      N <RETURN> {OFF}"
MD 6Ø PRINT"{2 DOWN}";SPC(11);
      :INPUT DT$
QA 7Ø PRINT"{CLR}{4 DOWN}
      {9 RIGHT}{RVS}{CYN} UPPE
      R AND LOWER CASE ? ":PRI
      NT:
RC 8Ø PRINT"{1Ø RIGHT}{RVS}
      {CYN} {WHT}Y{CYN} OR
      {WHT}N{CYN} THEN <RETURN
      > {OFF}"
CB 9Ø PRINT"{2 DOWN}";SPC(11);
      :INPUT CC$
PR 1ØØ IF CC$="Y" THEN CM$=CHR
       $(17):REM LOWER CASE
RC 11Ø IF CC$<>"Y" THEN CM$=CH
       R$(145):REM UPPER CASE
QF 12Ø PRINT"{CLR}{7}":POKE532
       8Ø,14:POKE53281,6
SX 13Ø PRINT"{4 DOWN}{RIGHT}
       {YEL}WHICH DISK DRIVE D
       O YOU WANT TO LIST?"
JC 14Ø PRINT"{2 DOWN}";SPC(11)
       ;:INPUT DI
MG 15Ø REM ***** READ DISK MEN
       U ******
XF 16Ø PRINT"{CLR}{CYN}":POKE5
       328Ø,2:POKE53281,Ø
HK 17Ø PRINT"{6 DOWN}{3 RIGHT}
       {RVS}{2 SPACES}READING
       {2 SPACES}DATA : PLEASE
        STANDBY{2 SPACES}"
ED 18Ø GOSUB163Ø
JD 19Ø OPEN8,DI,Ø,"$Ø":FORC=1T
       O8:GET#8,A$:NEXT:C=1:DN
       $="":FORC=1TO16
JA 2ØØ GET#B,A$:DN$=DN$+A$:NEX
       T:GET#B,A$:GET#8,A$:DN$
       =DN$+"{2 SPACES}":GET#8
       ,A$
ED 21Ø DN$=DN$+A$:GET#8,A$:DN$
       =DN$+A$:GET#8,A$:GET#8,
       A$
GE 22Ø GET#8,A$:GET#8,A$:C=1
FH 23Ø FORA=1TO4:GET#8,A$:NEXT
       :PN$="":TY$=""
PM 24Ø GET#8,A$:IFST<>ØTHEN31Ø
FP 25Ø IFA$=""THEN31Ø
MC 26Ø IFASC(A$)<>34THEN24Ø
BA 27Ø GET#8,A$:IFASC(A$)<>34T
       HENPN$=PN$+A$:GOTO27Ø
FA 2BØ GET#8,A$:IFASC(A$)=32TH
       EN28Ø
RR 29Ø TY$=TY$+A$:GET#8,A$:IFA
       $<>""THEN29Ø
RD 3ØØ TB$(C)=PN$:C=C+1:IFST=Ø
       THEN23Ø
XC 31Ø CLOSE8
GH 32Ø IF C>88 THEN GOSUB131Ø
JE 33Ø REM *** ALPHABETIZE LIS
       TING ***
CA 34Ø PRINT"{CLR}{CYN}":POKE5
       328Ø,4:POKE53281,Ø
QX 35Ø PRINT"{6 DOWN}{3 RIGHT}
       {RVS}{2 SPACES}SORTING
       {2 SPACES}DATA : PLEASE
        STANDBY{2 SPACES}"
JS 36Ø GOSUB157Ø
HA 37Ø Z$="ZZZZZZZZZZZZZZZ":E
       =1
GS 3BØ FORA=1TOC-1:C$=Z$:FORB=
       1TOC-1:IFC$<TB$(B)THEN4
       ØØ
JB 39Ø C$=TB$(B):D=B
QF 4ØØ NEXT:AB$(E)=C$:E=E+1:TB
       $(D)=Z$:NEXT
BM 41Ø REM{2 SPACES}**** JACKE
       T NAME = NS$ ***
MA 42Ø NS$="*****{3 SPACES}REF
       ERENCE{3 SPACES}*****"
SJ 43Ø REM ***** PRINT ALPHA L
       IST *****
AA 44Ø PRINT"{CLR}":POKE 532BØ
       ,5:POKE532B1,Ø
GR 45Ø PRINT"{6 DOWN}{2 RIGHT}
       {RVS}{2 SPACES}PRINTING
        JACKET : PLEASE STANDB
       Y{2 SPACES}"
MA 46Ø GOSUB151Ø
EP 47Ø DD=Ø:CD=INT(C/2):OPEN1,
       4
JS 4BØ FOR CR=1TO2
FS 49Ø PRINT#1,CHR$(1Ø):REM LI
       NEFEED
XA 5ØØ NEXT CR
JE 51Ø TL$="{O}"
FS 52Ø PRINT#1,TAB(2);:FOR TL=
       1 TO 72:PRINT#1,TL$;:NE
       XTTL:PRINT#1," CUT"
KC 53Ø GOSUB113Ø:GOSUB114Ø
RE 54Ø GOSUB113Ø
DC 55Ø PRINT#1,CHR$(14);NS$;CH
       R$(15);:REM 14 DOUBLE W
       IDTH 15 SINGLE
JG 56Ø GOSUB114Ø
PD 57Ø FOR LE=1TO2
RG 58Ø GOSUB113Ø:GOSUB114Ø
AD 59Ø NEXT LE
MK 6ØØ GOSUB113Ø
FX 61Ø PRINT#1,TAB(15);CM$;DN$
       ;SPC(5);DT$;:GOSUB114Ø:
       GOSUB116Ø
CJ 62Ø IF C>32 THEN79Ø
QA 63Ø REM ** PRINT : < 32 PRO
       GRAMS **
HC 64Ø FORDD=1TOCD:GOSUB113Ø
JE 65Ø PRINT#1,CHR$(16);CHR$(5
       Ø);CHR$(48);CM$;AB$(DD)
       ;:REM PRINT HEAD POSITI
       ON
KS 66Ø PRINT#1,CHR$(16);CHR$(5
       2);CHR$(53);CM$;AB$(CD+
```

Two Exciting New Books


COMPUTE!'s
FIRST BOOK of Commodore 128
Games, utilities, programming tips, and information for users of the Commodore 128 personal computer in 128 mode
A COMPUTE! Books Publication
$14.95

from COMPUTE!


Electronic Computer Projects
for Commodore and Atari Personal Computers
Soori Sivakumaran
A COMPUTE! Books Publication

COMPUTE!'s First Book of the Commodore 128

A spectacular collection of articles and programs exclusively for the Commodore 128 *in 128 mode.*

Edited

The editors at COMPUTE! Publications have collected some of the best games, programs, and tutorials for the Commodore 128 from *COMPUTE!* and *COMPUTE!'s Gazette*, plus some never-before-published articles and programs. Learn how to create windows, program sound, and make disks autoload. You'll even find a map of all the important memory locations. There's something for every 128 user. All programs run in 128 mode. *A disk is available which includes programs in the book, $12.95.*

$14.95 ISBN 0-87455-059-9

Electronic Computer Projects

Learn how to build all kinds of new devices to interface with your computer from inexpensive, available parts.

For the Commodore 64, 128, VIC, and any eight-bit Atari personal computer.

Soori Sivakumaran

This introduction to digital electronics and computer interfacing is the easy way to learn how computers interact with the outside world. Using a Commodore 64, 128, VIC, or any eight-bit Atari computer and *Electronic Computer Projects*, you'll be guided through the steps to building a joystick, light pen, game paddle, and numerous other devices. And since each project is independent from the others, you can choose only those projects that interest you. All the projects can be built at home and most require fewer than half a dozen parts.

$9.95 ISBN 0-87455-052-I

Visit your local book or computer store for these new titles. Or order directly from COMPUTE! Books. Call toll-free 800-346-6767 (in NY 212-887-8525) or write COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please include $2.00 per book ($5.00 air mail) for shipping and handling. NC residents add 4.5 percent sales tax. Allow 4–6 weeks from receipt of order for delivery.

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

```
          DD);:GOSU81140
MC 670  REM PRINT HEAD POSITION
AE 680  NEXT DD
KB 690  GOSU81130:GOSU81140:DD=
          DD+1
FR 700  IF DD>17 THEN720
JF 710  GOTO690
X8 720  GOSU81220
QF 730  FOR SL=1TO29
FK 740  GOSU81200:GOSUB1210
AE 750  NEXT SL
FE 760  GOSU81240
HG 770  GOSU81260
SE 780  REM ** PRINT : > 32 PRO
          GRAMS **
88 790  FORDD=1TO16:GOSUB1130
RX 800  PRINT#1,CHR$(16);CHR$(5
          0);CHR$(48);CM$;AB$(DD)
          ;:REM PRINT HEAD POSITI
          ON
QM 810  PRINT#1,CHR$(16);CHR$(5
          2);CHR$(53);CM$;AB$(DD+
          16);:GOSU81140
KF 820  REM PRINT HEAD POSITION
CR 830  NEXT DD
BB 840  GOSUB1130:GOSU81140
GJ 850  GOSU81220
HG 860  GOSUB1200:GOSUB1210
FM 870  CX=(C-33)/2:CZ=CX+32
QD 880  FORDD=33TOCZ:GOSU81200
8E 890  PRINT#1,CHR$(16);CHR$(5
          0);CHR$(48);CM$;AB$(DD)
          ;:REM PRINT HEAD POSITI
          ON
JQ 900  PRINT#1,CHR$(16);CHR$(5
          2);CHR$(48);CM$;AB$(DD+
          CX);:GOSUB1210
AD 910  REM PRINT HEAD POSITION
GC 920  NEXT DD
XH 930  GOSUB1200:GOSUB1210:DD=
          DD+1
JB 940  IF DD>60 THEN960
FD 950  GOTO930
AG 960  GOSU81240:GOSU81260
RG 970  REM *** CLOSING REMARKS
           ***
PA 980  PRINT"{CLR}{CYN}":POKE5
          3280,9:POKE53281,0
CC 990  PRINT"{6 DOWN}{2 RIGHT}
          {RVS}{2 SPACES}ALPHA8ET
          IZED DISK COVER COMPLET
          E{2 SPACES}"
MH 1000 GOSUB1460
QC 1010 PRINT"{5 DOWN}
          {2 RIGHT}{2 SPACES}DO
          {SPACE}YOU WANT ANOTHE
          R DISK COVER :"
RX 1020 INPUT"{6 RIGHT}'Y' OR
          {SPACE}'N' THEN <RETUR
          N>";AG$
FC 1030 FOR DD=0TO144:A8$(DD)=
          "":NEXTDD
BX 1040 IF AG$<>"Y"THEN1070
EP 1050 PRINT"{CLR}[7]":POKE53
          280,14:POKE53281,6:GOT
          O70
FP 1060 REM ** TERMINATE PROGR
          AM **
HC 1070 PRINT"{CLR}{CYN}":POKE
          53280,7:POKE53281,11
BP 1080 PRINT"{8 DOWN}
          {8 RIGHT}{WHT}{RVS}
          {2 SPACES}PROGRAM TERM
          INATED !{2 SPACES}"
FJ 1090 GOSUB1410
EJ 1100 FOR WT=1TO 1000:NEXT W
          T
GE 1110 PRINT"{CLR}[7]":POKE53
          280,14:POKE53281,6:END
RS 1120 REM *** DISK JACKET OU
          TLINE ***
```

```
SA 1130 PRINT#1,"{2 SPACES}[J]
          {7 SPACES}[G]*";:RETUR
          N
EM 1140 PRINT#1,CHR$(16);CHR$(
          54);CHR$(52);"*[M]
          {7 SPACES}[L]":RETURN
KE 1150 REM PRINT HEAD POSITIO
          N
J8 1160 PRINT#1,"{2 SPACES}[J]
          {7 SPACES}[G]*";
HQ 1170 PRINT#1,TA8(15);"
          [20 U]";:
JS 1180 PRINT#1,CHR$(16);CHR$(
          54);CHR$(52);"*[M]
          {7 SPACES}[L]":RETURN
F8 1190 REM PRINT HEAD POSITIO
          N
MP 1200 PRINT#1,"{10 SPACES}
          [J]";:RETURN:
CS 1210 PRINT#1,CHR$(16);CHR$(
          54);CHR$(53);"[L]":RET
          URN:REM PRINT HEAD POS
          ITION
GS 1220 CL$="C"
BD 1230 PRINT#1,TAB(2);:FOR CL
          =1 TO 72:PRINT#1,CL$;:
          NEXTCL:PRINT#1," FOLD"
          :RETURN
XR 1240 LL$="[U]"
DS 1250 PRINT#1,TAB(10);:FOR L
          L=1 TO 56:PRINT#1,LL$;
          :NEXTLL:PRINT#1," CUT"
          :RETURN
MR 1260 PRINT#1:FOR CR=1TO3
KM 1270 PRINT#1,CHR$(10)
DG 1280 NEXT CR:CLOSE1:GOTO980
DE 1290 REM ** MENU TOO LONG T
          O LIST **
FS 1300 REM ** CAN ONLY LIST 8
          8 PGMS **
HR 1310 PRINT"{CLR}{CYN}":POKE
          53280,1:POKE53281,7
8J 1320 PRINT"{5 DOWN}";SPC(10
          );"{CYN}TOO{2 SPACES}M
          ANY{2 SPACES}PROGRAMS"
CA 1330 PRINT SPC(10);"{CYN} T
          O LIST ON JACKET"
DR 1340 GOSUB1460
JG 1350 PRINT"{3 DOWN}";SPC(8)
          ;"{RED}PRINT{2 SPACES}
          THOSE{2 SPACES}THAT
          {2 SPACES}FIT?"
MC 1360 PRINT SPC(9);"({RVS}Y
          {OFF} OR {RVS}N{OFF} T
          HEN <RETURN>)
HE 1370 PRINT"{2 DOWN}";SPC(15
          );:INPUT AW$
FX 1380 IF AW$<>"Y" THEN1070
XR 1390 C=88:RETURN
SG 1400 REM{2 SPACES}**** SOUN
          D SUBROUTINES ***
SA 1410 REM{2 SPACES}##### 8UZ
          ZER #####
KR 1420 POKES,240
GF 1430 H=54273:S=54278:W=5427
          6:V=54296
BK 1440 POKEV,15:POKEH,5:POKEW
          ,33:FORT=0TO500:NEXT
CP 1450 FORT=H-1TOV:POKET,0:NE
          XT:RETURN
QM 1460 REM{2 SPACES}##### DON
          G #####
MG 1470 H=54273:S=54278:W=5427
          6:V=54296
QG 1480 POKES-1,9:POKEH,36:POK
          ES+9,16:POKEV,15:FORU=
          1TO4:POKEW,21:FORT=0TO
          500
EM 1490 NEXT:POKEW,20:NEXT
KJ 1500 FORT=H-1TOV:POKET,0:NE
          XT:RETURN
```

```
XF 1510 REM{2 SPACES}##### DIN
          G #####
MJ 1520 H=54273:S=54278:W=5427
          6:V=54296
XQ 1530 FOR AA=1TO3
CE 1540 POKEV,15:POKEH,40:POKE
          S-1,9:POKEW,17:FORT=1T
          O500:NEXTT
AK 1550 FORT=H-1TOV:POKET,0:NE
          XT
SX 1560 NEXTAA:RETURN
JD 1570 REM{2 SPACES}##### 8IN
          G-BONG #####
JX 1580 H=54273:S=54278:W=5427
          6:V=54296
DE 1590 POKEV,15:POKES-1,88:PO
          KES,89:POKEW-1,1:FORU=
          1TO6:POKEW,65
XR 1600 POKEH,20:FORT=0TO120:N
          EXT
BG 1610 POKEW,64:POKEH,50:POKE
          W,65:FORT=0TO120:NEXT:
          POKEW,64:NEXT
KA 1620 FORT=H-1TOV:POKET,0:NE
          XT:RETURN
JD 1630 REM{2 SPACES}##### BEL
          LS #####
XS 1640 V=54296:W=54276:POKEW+
          1,96
BQ 1650 POKEW+1,9
DM 1660 POKEV,15:FORL=1TO5:POK
          EW,21
PX 1670 POKEW-3,99*RND(1):POKE
          W+11,99*RND(1)
XK 1680 FORT=1TO600:NEXT:POKEW
          ,20:NEXT
XD 1690 FORI=W-4TOV:POKEI,0:NE
          XT:RETURN
```

Program 2: Jacket Lister for Atari 400, 800, XL, and XE

Version by Kevin Mykytyn, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
DC 10  DIM DATE$(12),UPPER$(1
       ),K$(1),DN$(10),DIR$(1
       7*88),T$(20),NAME$(26)
       ,SPC$(80)
FB 15  FOR A=1 TO 80:SPC$(A,A
       )=" ":NEXT A:NAME$="**
       *** REFERENCE DISK ***
       **":REM THIS MUST BE 2
       6 CHARACTERS
JM 20  OPEN #4,4,0,"K:"
KD 100  GRAPHICS 0:POKE 710,1
        5:POKE 709,0:POKE 712
        ,55
FD 110  POSITION 10,6:PRINT "
         WHAT IS TODAY'S DATE?
         "
JG 120  POSITION 6,8:PRINT "
        ENTER MO/DY/YR THEN
        RETURN "
PH 130  POSITION 15,13:INPUT
        DATE$:PRINT "{CLEAR}"
        :POKE 752,1
GA 160  PRINT "{CLEAR}":POSIT
        ION 2,5:PRINT "WHICH
        DISK DRIVE DO YOU WAN
        T TO LIST":POSITION 1
        7,7:PRINT "(1-9)"
FE 170  GOSUB 1000:IF K$<"1"
        OR K$>"9" THEN 170
HD 180  DN$="D1:*.*":DN$(2,2)
        =K$
JP 190  PRINT "{CLEAR}":POSIT
        ION 3,8:PRINT "[illegible]
       ING DATA : PLEASE STA
       NDBY ":POKE 712,136
IF 2ØØ TRAP 22Ø:FILE=1:OPEN
       #1,6,Ø,DN$
FM 21Ø INPUT #1,T$:DIR$((FIL
       E-1)*17+1,FILE*17)=T$
       :FILE=FILE+1:IF FILE<
       9Ø THEN 21Ø
MJ 22Ø FILE=FILE-2:TRAP 65ØØ
       Ø:CLOSE #1:IF PEEK(19
       5)=136 OR FILE=88 THE
       N 25Ø
LG 23Ø PRINT "{CLEAR}":POSIT
       ION 12,11:PRINT "DISK
        ERROR #";PEEK(195)
PN 24Ø POSITION 8,13:PRINT "
       PRESS ANY KEY TO RETR
       Y":GOSUB 1ØØØ:GOTO 19
       Ø
BH 25Ø POSITION 5,8:PRINT "S
       ORTING":POKE 712,2ØØ
MJ 26Ø G=INT(FILE/2)
6H 265 N=Ø:FOR I=1 TO FILE-G
CC 27Ø IF DIR$((I-1)*17+1,I*
       17)<=DIR$((I+G-1)*17+
       1,(I+G)*17) THEN 29Ø
BJ 28Ø T$=DIR$((I-1)*17+1,I*
       17):DIR$((I-1)*17+1,I
       *17)=DIR$((I+G-1)*17+
       1,(I+G)*17):DIR$((I+G
       -1)*17+1,(I+G)*17)=T$
       :N=1
HE 29Ø NEXT I:IF N=1 THEN 26
       5
ID 3ØØ G=INT(G/2):IF G>=1 TH
       EN 265
AD 31Ø POSITION 2,8:PRINT "
        PRINTING JACKET : PL
       EASE STANDBY ":POKE
       712,1Ø4:OPEN #1,4,4,"
       P:"
MH 32Ø FOR A=1 TO 3:PRINT #1
       ;CHR$(13):NEXT A
6G 33Ø GOSUB 3Ø7Ø:PRINT #1;"
        CUT"
NH 34Ø GOSUB 3ØØØ:GOSUB 3Ø1Ø
       :GOSUB 3Ø2Ø
ND 35Ø GOSUB 3ØØØ:PRINT #1;C
       HR$(14);NAME$;CHR$(2Ø
       );:GOSUB 3Ø2Ø:REM 14
       IS DOUBLE WIDTH, 2Ø I
       S NORMAL WIDTH
DI 36Ø FOR A=1 TO 2:GOSUB 3Ø
       ØØ:GOSUB 3Ø1Ø:GOSUB 3
       Ø2Ø:NEXT A
HL 37Ø GOSUB 3ØØØ:PRINT #1;S
       PC$(1,22);DATE$;SPC$(
       1,3Ø-LEN(DATE$));:GOS
       UB 3Ø2Ø
HD 38Ø CD=INT(FILE/2):IF FIL
       E>32 THEN 48Ø
CA 39Ø FOR DD=1 TO CD:GOSUB
       3ØØØ
JK 4ØØ PRINT #1;SPC$(1,8);DI
       R$((DD-1)*17+1,DD*17-
       3);SPC$(1,9);DIR$((CD
       +DD-1)*17+1,(CD+DD)*1
       7-3);SPC$(1,7);
NL 41Ø GOSUB 3Ø2Ø:NEXT DD
LJ 42Ø GOSUB 3ØØØ:GOSUB 3Ø1Ø
       :GOSUB 3Ø2Ø:DD=DD+1
BH 43Ø IF DD>17 THEN 45Ø
6H 44Ø GOTO 42Ø
KC 45Ø GOSUB 3Ø7Ø:PRINT #1;"
        FOLD"
CJ 46Ø FOR SL=1 TO 29:GOSUB
       3Ø5Ø:GOSUB 3Ø6Ø:GOSUB
        3Ø65
LK 47Ø NEXT SL:GOSUB 3Ø9Ø:PR
       INT #1;" CUT":GOTO 2Ø
       ØØ
AA 48Ø FOR DD=1 TO 16:GOSUB
       3ØØØ
```

```
6D 49Ø PRINT #1;SPC$(1,8);DI
       R$((DD-1)*17+1,DD*17-
       3);SPC$(1,9);DIR$((16
       +DD-1)*17+1,(16+DD)*1
       7-3);SPC$(1,7);
NL 5ØØ GOSUB 3Ø2Ø:NEXT DD
NG 51Ø GOSUB 3ØØØ:GOSUB 3Ø1Ø
       :GOSUB 3Ø2Ø
CN 52Ø GOSUB 3Ø7Ø:PRINT #1;"
        FOLD":GOSUB 3Ø5Ø:GOS
       UB 3Ø6Ø:GOSUB 3Ø65
FA 53Ø CX=INT((FILE-33)/2):C
       Z=CX+32
6N 54Ø FOR DD=33 TO CZ:GOSUB
        3Ø5Ø
F6 55Ø PRINT #1;SPC$(1,9);DI
       R$((DD-1)*17+1,DD*17-
       3);SPC$(1,6);DIR$((DD
       +CX-1)*17+1,(DD+CX)*1
       7-3);SPC$(1,11);"!"
6C 56Ø NEXT DD
NC 57Ø GOSUB 3Ø5Ø:GOSUB 3Ø6Ø
       :GOSUB 3Ø65:DD=DD+1
CC 58Ø IF DD<61 THEN 57Ø
KF 59Ø GOSUB 3Ø9Ø:PRINT #1;"
        CUT":GOTO 2ØØØ
HA 1ØØØ POKE 752,1:GET #4,K:
        K$=CHR$(K):RETURN
KA 2ØØØ FOR CR=1 TO 3:PRINT
        #1;CHR$(13):NEXT CR:
        CLOSE #1
PL 2Ø1Ø PRINT "{CLEAR}":POSI
        TION 3,1Ø:PRINT "ALP
        HABETIZED DISK COVER
         COMPLETE"
ML 2Ø2Ø POSITION 3,13:PRINT
        "DO YOU WANT ANOTHER
         DISK COVER?"
DM 2Ø3Ø GOSUB 1ØØØ:IF K$="Y"
         THEN RUN
ME 2Ø4Ø IF K$<>"N" THEN 2Ø3Ø
KO 2Ø5Ø PRINT "{CLEAR}
        {2 DOWN}BYE":POKE 75
        2,Ø:END
DE 3ØØØ PRINT #1;" !
        {8 SPACES}!*";:RETUR
        N
FE 3Ø1Ø PRINT #1;SPC$(1,52);
        :RETURN
KL 3Ø2Ø PRINT #1;"*!
        {8 SPACES}!":RETURN
JO 3Ø5Ø PRINT #1;"
        {1Ø SPACES}!";:RETURN
FL 3Ø6Ø PRINT #1;SPC$(1,54);
        :RETURN
6J 3Ø65 PRINT #1;"!":RETURN
BH 3Ø7Ø PRINT #1;"  ";:FOR A
        =1 TO 72:PRINT #1;"-
        ";:NEXT A:RETURN
BJ 3Ø9Ø PRINT #1;"
        {11 SPACES}";:FOR A=1
         TO 54:PRINT #1;"-";
        :NEXT A:RETURN
```

Program 3: Apple II Jacket Lister

Version by Tim Midkiff, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
79 1Ø REM ***** PROGRAM SET UP *
      ****
6D 2Ø DIM TB$(144),AB$(144),WS(1
      ØØØ)
CA 3Ø HOME : PRINT : PRINT "WHAT
       IS TODAY'S DATE (MO/DY/YR
      )";: INPUT DT$
5F 4Ø HOME : PRINT : PRINT "WHIC
      H DRIVE DO YOU WANT TO LIS
      T (1/2)";: INPUT D1
AC 5Ø IF D1 < 1 OR D1 > 2 THEN 4
      Ø
2B 6Ø REM ***** READ DISK MENU *
      ****
37 7Ø GOSUB 132Ø: HOME : PRINT "
      READING DATA : PLEASE STAN
      DBY"
78 8Ø FOR I = 768 TO 779: READ A
      : POKE I,A: NEXT :P1 = Ø:P
      2 = Ø:A$ = "":C = Ø
6Ø 9Ø P1 = WS(Ø) - WS(Ø) + PEEK
      (131):P2 = WS(Ø) - WS(Ø) +
      PEEK (132)
34 1ØØ POKE 769,P1: POKE 77Ø,P2
AE 11Ø POKE 54,Ø: POKE 55,3: POK
       E 56,11: POKE 57,3: CALL
       1ØØ2
9A 12Ø PRINT CHR$ (4);"CATALOG,D
       ";D1
F8 125 PRINT
15 13Ø POKE 768,173: POKE 769,P1
       : POKE 77Ø,P2
4Ø 14Ø POKE 54,11: POKE 55,3: PO
       KE 56,Ø: POKE 57,3: CALL
       1ØØ2
71 15Ø FOR I = 1 TO 4: INPUT A$:
       NEXT :C = 1
B8 16Ø INPUT A$: IF A$ = "" THEN
       17Ø
C2 165 IF LEFT$ (A$,1) = "*" THE
       N A$ = RIGHT$ (A$, LEN (A
       $) - 1)
38 167 TB$(C) = MID$ (A$,7,18):C
       = C + 1: GOTO 16Ø
D7 17Ø POKE 54,24Ø: POKE 55,253:
       POKE 56,27: POKE 57,253:
       CALL 1ØØ2
3C 18Ø FOR I = 1 TO C - 1: PRINT
       TB$(I): NEXT
71 19Ø DATA 141,Ø,64,238,1,3,2Ø8
       ,3
EA 2ØØ DATA 238,2,3,96
CF 34Ø IF C > 88 THEN GOSUB 126Ø
12 35Ø REM *** ALPHABETIZE LISTI
       NG ***
F9 36Ø GOSUB 132Ø: HOME : PRINT
       "SORTING DATA : PLEASE ST
       ANDBY"
4A 37Ø Z$ = CHR$ (255):E = 1
7D 38Ø FOR A = 1 TO C - 1:C$ = Z
       $: FOR B = 1 TO C - 1: IF
       C$ < TB$(B) THEN 4ØØ
84 39Ø C$ = TB$(B):D = B
52 4ØØ NEXT :AB$(E) = C$:E = E +
       1:TB$(D) = Z$: NEXT
87 41Ø REM **** JACKET NAME = NS
       $ ****
6D 42Ø NS$ = "*****   REFERENCE
       *****"
62 43Ø REM ***** PRINT ALPHA LIS
       T *****
CB 44Ø GOSUB 132Ø: HOME : PRINT
       "PRINTING JACKET : PLEASE
       STANDBY"
A2 45Ø DD = Ø:CD = INT (C / 2):
       PRINT CHR$ (4);"PR#1": PR
       INT CHR$ (9);"8ØN"
EA 46Ø FOR CR = 1 TO 2
8B 47Ø PRINT CHR$ (1Ø): REM LINE
       FEED
8C 48Ø NEXT
97 51Ø TL$ = "-"
7A 52Ø PRINT TAB( 4);: FOR TL =
       1 TO 71: PRINT TL$;: NEXT
       : PRINT " CUT"
2D 53Ø GOSUB 11ØØ: GOSUB 111Ø
CB 54Ø GOSUB 11ØØ
D6 55Ø POKE 36, INT (4Ø - LEN (N
       S$) / 2): PRINT NS$;
D3 56Ø GOSUB 111Ø
A9 57Ø FOR LE = 1 TO 2
```

```
37 58Ø GOSUB 11ØØ: GOSUB 111Ø
ØF 59Ø NEXT
C4 6ØØ GOSUB 11ØØ
01 61Ø POKE 36,36: PRINT OT$;: G
       OSUB 111Ø: GOSUB 112Ø
F3 62Ø IF C > 32 THEN 79Ø
A6 63Ø REM *** PRINT : < 32 PROG
       RAMS ***
19 64Ø FOR OO = 1 TO CO: GOSUB 1
       1ØØ
96 65Ø POKE 36,2Ø: PRINT AB$(OO)
       ;
BB 66Ø POKE 36,45: PRINT AB$(CO
       + OO);
D6 67Ø GOSUB 111Ø
ØE 68Ø NEXT
B5 69Ø GOSUB 11ØØ: GOSUB 111Ø:OO
        = OO + 1
AB 7ØØ IF OO > 17 THEN 72Ø
1E 71Ø GOTO 69Ø
E5 72Ø GOSUB 117Ø
2Ø 73Ø FOR SL = 1 TO 29
95 74Ø GOSUB 115Ø: GOSUB 116Ø
Ø9 75Ø NEXT
F5 76Ø GOSUB 119Ø
D9 77Ø GOSUB 121Ø
B2 7BØ REM *** PRINT : > 32 PROG
       RAMS ***
41 79Ø FOR OO = 1 TO 16: GOSUB 1
       1ØØ
BE 8ØØ POKE 36,2Ø: PRINT AB$(OO)
       ;
92 B1Ø POKE 36,45: PRINT AB$(OO
       + 16);
CE B2Ø GOSUB 111Ø
Ø6 B3Ø NEXT
32 B4Ø GOSUB 11ØØ: GOSUB 111Ø
EC 85Ø GOSUB 117Ø
9A 86Ø GOSUB 115Ø: GOSUB 116Ø
77 87Ø CX = (C - 33) / 2:CZ = CX
        + 32
F1 BBØ FOR OO = 33 TO CZ: GOSUB
       115Ø
AØ 89Ø POKE 36,2Ø: PRINT A8$(OO)
       ;
16 9ØØ POKE 36,4Ø: PRINT AB$(OO
       + CX);
E1 91Ø GOSUB 116Ø
Ø5 92Ø NEXT
11 93Ø GOSUB 115Ø: GOSUB 116Ø:OO
        = OD + 1
2C 94Ø IF OO > 6Ø THEN 96Ø
A3 95Ø GOTO 93Ø
63 96Ø GOSUB 119Ø: GOSUB 121Ø
1F 97Ø REM *** CLOSING REMARKS *
       **
34 9BØ PRINT CHR$ (4);"PR#Ø"
3Ø 99Ø GOSUB 132Ø: HOME : PRINT
       "ALPHABETIZEO OISK COVER
       COMPLETE"
13 1ØØØ PRINT : PRINT "OO YOU WA
       NT ANOTHER OISK COVER (Y
       /N)";: INPUT AG$
77 1Ø1Ø FOR OO = Ø TO 144:AB$(OO
       ) = "": NEXT : RESTORE
ØB 1Ø2Ø IF AG$ < > "Y" THEN 1Ø5Ø
11 1Ø3Ø GOTO 4Ø
5A 1Ø4Ø REM ** TERMINATE PROGRAM
        **
2A 1Ø5Ø HOME : PRINT "PROGRAM TE
       RMINATEO"
5F 1Ø6Ø GOSUB 132Ø
AB 1Ø7Ø FOR WT = 1 TO 1ØØØ: NEXT
FF 1ØBØ HOME : ENO
1A 1Ø9Ø REM *** OISK JACKET OUTL
       INE ***
FE 11ØØ PRINT "  !         ! *";:
       RETURN
F9 111Ø POKE 36,64: PRINT "* !
             !": RETURN
42 112Ø PRINT "  !         ! *";
FC 113Ø POKE 36,29: PRINT "-----
       ---------------";
Ø6 114Ø POKE 36,64: PRINT "* !
             !": RETURN
55 115Ø PRINT "             !";: RE
       TURN
DC 116Ø POKE 36,66: PRINT "!": R
       ETURN
DF 117Ø CL$ = "-"
6E 118Ø PRINT TAB( 4);: FOR CL =
        1 TO 71: PRINT CL$;: NE
       XT : PRINT " FOLO": RETU
       RN
7B 119Ø LL$ = "-"
03 12ØØ PRINT TAB( 12);: FOR LL
       = 1 TO 55: PRINT LL$;: N
       EXT : PRINT " CUT": RETU
       RN
F9 121Ø PRINT : FOR CR = 1 TO 3
BØ 122Ø PRINT CHR$ (1Ø): REM LIN
       EFEEO
9C 123Ø NEXT : GOTO 98Ø
F2 124Ø REM *** MENU TOO LONG TO
        LIST ***
7C 125Ø REM *** CAN ONLY LIST BB
        PGMS ***
1F 126Ø GOSUB 132Ø: HOME : PRINT
        "TOO MANY PROGRAMS TO L
       IST ON JACKET"
F6 127Ø PRINT : PRINT "PRINT THO
       SE THAT FIT (Y/N)";: INP
       UT AW$
2C 128Ø IF AW$ < > "Y" THEN 1Ø5Ø
BB 129Ø C = 8B: RETURN
47 13ØØ RE
48 131Ø REM ***** SOUNO ROUTINE
       *****
AB 132Ø FOR I = 1 TO 1Ø:A = PEEK
        ( - 16336): NEXT : RETU
       RN
```

Program 4: ProDOS Modifications for Program 3

Refer to the article for Instructions on odding these replacement lines.

```
B3 8Ø O$ = CHR$ (4): PRINT O$;"P
      REFIX,D";O1: PRINT D$;"PRE
      FIX"
C2 9Ø INPUT P$
C3 1ØØ PRINT O$;"OPEN ";P$;",TOI
       R"
3Ø 11Ø PRINT O$;"REAO ";P$
63 12Ø FOR I = 1 TO 3: INPUT A$:
        NEXT :C = 1
DC 13Ø INPUT A$: IF LEN (A$) > Ø
        THEN TB$(C) = MIO$ (A$,2
       ,15): PRINT TB$(C):C = C
       + 1: GOTO 13Ø
79 14Ø PRINT O$;"CLOSE ";P$
```

Program 5: IBM PC/PCjr Jacket Lister

Version by Tim Midkiff, Editorial Programmer

For Instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" In this Issue of COMPUTE!.

```
OF 1Ø KEY OFF:WIOTH 8Ø:OEF SEG=Ø
      :POKE 1Ø47,PEEK(1Ø47) OR 6
      4
OB 2Ø OIM T8$(144):OIM A8$(144)
EJ 3Ø CLS:PRINT:PRINT "What is t
      oday's date (Mo/Oy/Yr)";:I
      NPUT OT$
6F 4Ø CLS:PRINT:PRINT "Which dis
      k drive do you want to lis
      t (A/B)";:INPUT OI$:IF OI$
      <>"A" ANO OI$<>"B" THEN 4Ø
L6 5Ø REM *** REAO OISK MENU ***
KL 6Ø BEEP:CLS:PRINT "REAOING OA
      TA : PLEASE STANDBY"
CP 7Ø FSPEC$=OI$+":*.*"
E6 BØ HEAO=1Ø5Ø:TAIL=1Ø52:BUFFER
      =1Ø54:C=Ø
LK 9Ø ON ERROR GOTO 11Ø
OC 1ØØ FILES FSPEC$:ON ERROR GOT
       O Ø:GOTO 12Ø
IN 11Ø BEEP:CLS:PRINT "CANNOT RE
       AO OIRECTORY":ON ERROR GO
       TO Ø:END
6K 12Ø OIM TT$(24):LOCATE 3,1:RO
       WS=Ø
OH 13Ø POKE HEAO,3Ø:POKE TAIL,34
       :POKE BUFFER,Ø:POKE BUFFE
       R+1,79:POKE 8UFFER+2,13:P
       OKE 8UFFER+3,2B
HC 14Ø LINE INPUT TT$(ROWS):IF T
       T$(ROWS)<>"" THEN ROWS=RO
       WS+1:GOTO 13Ø
EH 15Ø ROWS=ROWS-1:FOR I=Ø TO RO
       WS:FOR J=Ø TO 3
6B 16Ø T$=MIO$(TT$(I),J*1B+1,12)
KC 17Ø IF T$<>"" THEN TB$(C)=T$:
       C=C+1
PK 18Ø NEXT J:NEXT I:ERASE TT$
CD 19Ø IF C>BB THEN GOSU8 126Ø
CL 2ØØ REM *** ALPHABETIZE LISTI
       NG ***
EB 21Ø BEEP:CLS:PRINT "SORTING O
       ATA : PLEASE STANDOY"
FF 22Ø Z$=CHR$(255):E=1
QK 23Ø FOR A=Ø TO C-1:C$=Z$:FOR
       B=Ø TO C-1:IF C$<TB$(B) T
       HEN 25Ø
8F 24Ø C$=T8$(8):O=8
FE 25Ø NEXT:A8$(E)=C$:E=E+1:TB$(
       D)=Z$:NEXT
IE 41Ø REM *** JACKET NAME = NS$
        ***
CN 42Ø NS$="*****  REFERENCE  **
       ***"
PO 43Ø REM *** PRINT ALPHA LIST
       ***
6O 44Ø BEEP:CLS:PRINT "PRINTING
       JACKET : PLEASE STANOBY"
EF 45Ø OO=Ø:CO=INT(C/2)
NL 46Ø FOR CR=1 TO 2
CA 47Ø LPRINT CHR$(1Ø):REM LINEF
       EEO
CE 4BØ NEXT CR
GB 51Ø TL$="-"
DC 52Ø LPRINT TA8(3);:FOR TL=1 T
       O 71:LPRINT TL$;:NEXT TL:
       LPRINT " CUT"
MF 53Ø GOSU8 11ØØ:GOSU8 111Ø
OA 54Ø GOSUB 11ØØ
IA 55Ø LPRINT TAB(INT(39-LEN(NS$
       )/2));NS$;
PB 56Ø GOSUB 111Ø
KN 57Ø FOR LE=1 TO 2
NP 5BØ GOSUB 11ØØ:GOSUB 111Ø
PC 59Ø NEXT LE
OJ 6ØØ GOSUB 11ØØ
EA 61Ø LPRINT TA8(36);CM$;OT$;:G
       OSUB 111Ø:GOSUB 112Ø
KN 62Ø IF C>32 THEN 79Ø
BM 63Ø REM *** PRINT : < 32 PROG
       RAMS ***
HO 64Ø FOR OO=1 TO CO:GOSUB 11ØØ
FB 65Ø LPRINT TAB(2Ø);AB$(OO);
N6 66Ø LPRINT TA8(45);AB$(CO+OO)
       ;
PE 67Ø GOSUB 111Ø
J6 68Ø NEXT OO
CA 69Ø GOSU8 11ØØ:GOSU8 111Ø:OO=
       OO+1
FC 7ØØ IF OO>17 THEN 72Ø
KJ 71Ø GOTO 69Ø
EJ 72Ø GOSUB 117Ø
CO 73Ø FOR SL=1 TO 29
IC 74Ø GOSU8 115Ø:GOSUB 116Ø
```

```
IP 750 NEXT SL
GL 760 GDSUB 1190
AB 770 GOSUB 1210
EB 7B0 REM *** PRINT : > 32 PRDG
       RAMS ***
FL 790 FDR DD=1 TD 16:GDSUB 1100
FJ B00 LPRINT TAB(20);AB$(DD);
GN B10 LPRINT TAB(45);AB$(DD+16)
       ;
PM B20 GDSUB 1110
JO B30 NEXT DD
NK B40 GDSUB 1100:GDSUB 1110
EA B50 GDSUB 1170
IH B60 GDSUB 1150:GDSUB 1160
KA B70 CX=(C-33)/2:CZ=CX+32
SA BB0 FDR DD=33 TO CZ:GDSUB 115
       0
SL B90 LPRINT TAB(20);AB$(DD);
QK 900 LPRINT TAB(45);AB$(DD+CX)
       ;
DM 910 GDSUB 1160
JN 920 NEXT DD
NA 930 GDSUB 1150:GOSUB 1160:DD=
       DD+1
LF 940 IF DD>60 THEN 960
IP 950 GDTO 930
SL 960 GDSUB 1190:GDSUB 1210
EA 970 REM *** CLDSING REMARKS *
       **
DM 9B0 BEEP:CLS:PRINT "ALPHABETI
       ZED DISK COVER CDMPLETE"
PI 990 PRINT:PRINT "Do you want
       another disk cover (Y/N)"
       ;:INPUT AG$
IN 1000 FDR DD=0 TO 144:AB$(DD)=
        "":NEXT DD
LN 1010 IF AG$<>"Y" THEN 1040
MJ 1020 CLS:GOTD 40
QK 1030 REM ** TERMINATE PRDGRAM
         **
HF 1040 BEEP:CLS:PRINT "PRDGRAM
        TERMINATED"
NE 1050 FOR WT=1 TD 1000:NEXT WT
BH 1060 CLS:END
OA 1090 REM *** DISK JACKET DUTL
        INE ***
IP 1100 LPRINT " !          ! *";:R
        ETURN
DC 1110 LPRINT TAB(64);"* !
          !":RETURN
BD 1120 LPRINT " !          ! *";
QN 1130 LPRINT TAB(29);"---------
        -------------";
EL 1140 LPRINT TAB(64);"* !
          !":RETURN
CC 1150 LPRINT "            !";:RET
        URN
KJ 1160 LPRINT TAB(66);"!":RETUR
        N
GO 1170 CL$="-"
PJ 11B0 LPRINT TAB(3);:FDR CL=1
        TO 71:LPRINT CL$;:NEXT C
        L:LPRINT " FOLD":RETURN
KK 1190 LL$="-"
OF 1200 LPRINT TAB(11);:FDR LL=1
         TD 55:LPRINT LL$;:NEXT
        LL:LPRINT" CUT":RETURN
DN 1210 LPRINT:FDR CR=1 TD 3
CP 1220 LPRINT CHR$(10):REM LINE
        FEED
LB 1230 NEXT CR:GDTD 9B0
OF 1240 REM ** MENU TDD LDNG TD
        LIST **
IE 1250 REM ** CAN DNLY LIST BB
        PGMS **
OK 1260 BEEP:CLS:PRINT "TDD MANY
         PRDGRAMS TD LIST DN JAC
        KET"
CC 1270 PRINT:PRINT "Print those
         that fit (Y/N)";:INPUT
        AW$
GG 12B0 IF AW$<>"Y" THEN 1040
OO 1290 C=BB:RETURN
```

©

64 Encryptor

James Pettus

This BASIC utility will hide your programs from prying eyes. It encrypts a BASIC program in memory so that it can be neither stopped while running nor listed. The program also includes an option for restoring things back to normal if you wish. A secret ID code even prevents people who have the Encryptor program themselves from unlocking your secrets.

Part of the fun of computing is sharing one of your programs with others. At times, however, you may want to keep things confidential. For example, you might have written a finance program which contains DATA statements revealing your entire personal portfolio. You might want to prevent others from looking at this information. The LIST command ordinarily displays the contents of any BASIC program.

However, you can use "64 Encryptor" to encrypt any BASIC program to prevent other people from deciphering it. Though the encrypted program can't be listed or examined, it still runs normally. And since each copy of Encryptor has a unique ID code, your protected program should be safe even from others who have 64 Encryptor themselves.

A Special Random Identifier

Type in and save the BASIC loader program listed below. You may save it with any filename you like, except ENCRYPTOR (that's what the BASIC loader will name the machine language file that it creates). When the program runs, it spends a few seconds creating the Encryptor machine language routine in the memory area starting at 49152, then it saves the machine language to disk. To have the Encryptor file saved to tape instead, change the DV=8 in line 80 to DV=1.

When the loader writes Encryptor into high memory, it embeds an identifier mark within the program. The identifier is randomly selected and will be different each time you run the loader. This feature makes a program encrypted with one copy of Encryptor incompatible with any other copy of Encryptor—even another copy created on the same 64. As a result, you don't have to worry that other people with this program can decrypt your programs.

To encrypt or decrypt a BASIC program, follow these steps:

- Load Encryptor with LOAD"ENCRYPTOR",8,1 for disk or LOAD "ENCRYPTOR",1,1 for tape.

- Type NEW and press RETURN.
- Load the BASIC program you wish to encrypt or decrypt.
- To encrypt a program, type SYS 49152 and press RETURN. When the cursor returns, be sure to immediately save a copy of the encrypted version using a different filename.
- To decrypt a program, type SYS 49155 and press RETURN.

An encrypted program runs normally, but cannot easily be examined by the person using it. When you run an encrypted program, a built-in machine language subroutine is called to decrypt the actual program data and run it. At the same time, Encryptor disables the LIST command and the RUN/STOP–RESTORE key combination. You should make sure that the program being encrypted does not contain any references to the ROM routine at 65505 ($FFE1), which tests to see whether the RUN/STOP key has been pressed. The program to be protected also should not offer the user the option of exiting the program.

Because the BASIC loader program creates a different Encryptor each time it is run, you should take care to make a backup copy of each Encryptor that you create. (You should also keep an unprotected copy of any important programs you encrypt.) If you accidentally erase your only copy of Encryptor, you will not be able to decrypt any programs protected with that version. Of course, to keep your programs secure, you should not give anyone else a copy of your version of Encryptor.

Works With BASIC/ Machine Language Combinations

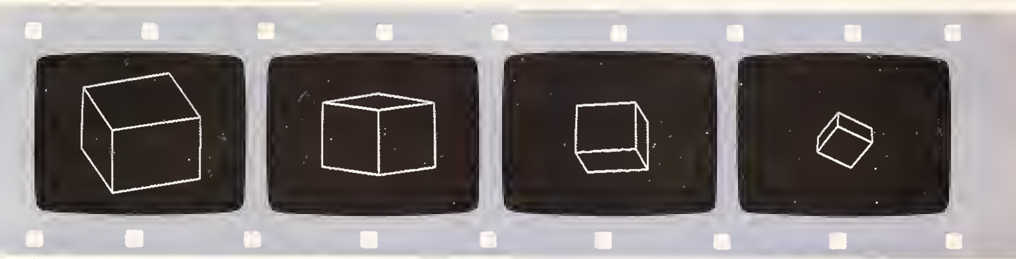
Some BASIC programs require that you relocate the start of BASIC text before you load and run them, others leave little memory for variables (meaning you should not enlarge the program), and some BASIC programs cannot be relocated because they have ML routines appended to the end of BASIC text. Encryptor has been designed with all these conditions in mind. The ML routine included in an encrypted program contains no absolute addresses, and it moves program data down in memory after it has done its work, so nonrelocating BASIC programs can still be safely encrypted.

64 Encryptor

For instructions on entering this listing. please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
GH 10 PRINTCHR$(147)CHR$(155)"
      PLEASE WAIT":I=49152
HG 20 READA:IFA=256THEN40
HJ 30 POKEI,A:CK=CK+A:I=I+1:GO
      TO20
MC 40 IFCK<>66816THENPRINT"ERR
      OR IN DATA STATEMENTS.":
      STOP
JG 50 POKEI+4574,255
FQ 60 POKEI+4577,128:FORA=I-31
      7TOI-308:POKEA,PEEK(I+45
      86):NEXT:POKEI+4577,0
DB 70 POKEI+4574,0
H8 80 DV=8:SYS57812"ENCRYPTOR"
      ,DV
XG 90 PRINT"SAVING ENCRYPTOR"
JM 100 POKE251,0:POKE252,192:P
       OKE780,251:POKE782,I/25
       6:POKE781,I-PEEK(782)*2
       56
PH 110 SYS65496:PRINT"ENCRYPTO
       R CREATED.":END
EE 120 DATA76,254,192,76,28,19
       3,167,43
RH 130 DATA135,251,135,253,167
       ,44,135,254
EE 140 DATA232,134,252,160,0,1
       77,251,145
AF 150 DATA253,230,251,230,253
       ,208,4,230
KH 160 DATA252,230,254,167,252
       ,197,46,208
SQ 170 DATA236,167,251,197,45,
       208,230,198
RF 180 DATA46,96,167,44,135,17
       5,167,43
AM 190 DATA56,233,1,176,2,198,
       175,133
SF 200 DATA174,167,46,135,252,
       232,134,254
AJ 210 DATA167,45,56,233,1,176
       ,4,198
GC 220 DATA252,198,254,133,251
       ,133,253,160
HC 230 DATA0,177,251,145,253,1
       98,251,198
GA 240 DATA253,167,251,201,255
       ,208,4,198
MB 250 DATA252,198,254,167,252
       ,197,175,208
KC 260 DATA232,167,251,197,174
       ,208,226,230
QH 270 DATA46,96,167,43,135,25
       1,167,44
RJ 280 DATA232,134,252,160,0,1
       62,8,177
FR 290 DATA251,10,102,255,202,
       208,250,167
EQ 300 DATA255,145,251,230,251
       ,208,2,230
EQ 310 DATA252,167,252,197,46,
       208,230,167
CD 320 DATA251,197,45,208,224,
       96,169,0
CX 330 DATA133,255,160,165,191
       ,79,192,69
RC 340 DATA255,133,255,209,43,
       208,6,200
P8 350 DATA192,175,208,240,96,
       169,199,160
QX 360 DATA192,32,30,171,108,2
       ,160,69
KP 370 DATA78,67,82,89,80,84,7
       9,82
DQ 380 DATA32,73,46,68,46,32,7
       7,73
AK 390 DATA83,77,65,84,67,72,0
       ,169
DR 400 DATA0,133,255,160,165,1
       91,79,192
GM 410 DATA69,255,133,255,145,
       43,200,192
CX 420 DATA175,208,242,96,0,0,
       0,0
GC 430 DATA0,0,0,0,0,0,32,50
JC 440 DATA192,32,122,192,160,
       0,191,48
GG 450 DATA193,145,43,200,208,
       248,32,223
QM 460 DATA192,32,89,166,32,51
       ,165,104
FD 470 DATA104,108,2,160,32,16
       6,192,32
EA 480 DATA122,192,32,6,192,32
       ,89,166
AC 490 DATA32,51,165,104,104,1
       08,2,160
RR 500 DATA25,8,0,0,158,194,40
       ,52
MS 510 DATA51,41,170,50,53,54,
       172,194
JA 520 DATA40,52,52,41,170,50,
       54,0
XF 530 DATA0,0,167,43,135,251,
       167,44
BJ 540 DATA232,134,252,160,0,1
       62,8,177
FR 550 DATA251,10,102,255,202,
       208,250,167
PR 560 DATA255,145,251,230,251
       ,208,2,230
HR 570 DATA252,167,252,197,46,
       208,230,167
JR 580 DATA251,197,45,208,224,
       160,84,177
MQ 590 DATA43,153,172,1,200,19
       2,165,208
AH 600 DATA246,76,0,2,167,43,1
       35,251
DP 610 DATA135,253,167,44,135,
       254,232,134
RS 620 DATA252,160,0,177,251,1
       45,253,230
XB 630 DATA251,230,253,208,4,2
       30,252,230
PX 640 DATA254,167,252,197,46,
       208,236,167
DF 650 DATA251,197,45,208,230,
       198,46,32
PB 660 DATA89,166,32,51,165,12
       0,162,255
JJ 670 DATA169,182,143,6,3,169
       ,234,143
AG 680 DATA40,3,169,246,143,41
       ,3,169
FK 690 DATA193,143,24,3,169,25
       4,143,25
PG 700 DATA3,88,76,174,167,0,0
       ,0
RP 710 DATA0,0,0,0,0,0,0,0
KM 720 DATA0,0,0,0,0,0,0,0
FK 730 DATA0,0,0,0,0,0,0,0
BJ 740 DATA0,0,0,0,0,0,0,0
RJ 750 DATA0,0,0,0,0,0,0,0
KX 760 DATA0,0,0,0,0,0,0,0
F8 770 DATA0,0,0,0,0,0,0,0
BR 780 DATA0,0,0,0,0,0,0,0
RR 790 DATA0,0,0,0,0,0,0,0
MR 800 DATA0,0,0,0,0,0,0,0
GQ 810 DATA0,0,0,0,0,0,0,0
HF 820 DATA0,256
```

©



With ANIMATE you can create rapidly moving 3-D graphics within a BASIC program. This series of photos shows only 4 of the 95 screens used for the CUBE display, which creates a rotating cube that moves toward and away from the viewer.

Easy IBM Full-Screen Animation

Paul W. Carlson

Now you can write BASIC programs with smooth, flicker-free animated displays that move at machine language speeds. For the IBM PC/PCjr. BASICA and a color/graphics card are required to use the program on the PC. Cartridge BASIC is required for the PCjr.

Full-screen animation is achieved by rapidly displaying a series of high-resolution screens on the video display. Producing realistic animation using BASIC is very difficult because of the time required to create the screen images. The creation of a high-resolution screen image usually consists of two processes repeated many times. First, the coordinates of the endpoints of a line segment are computed. Second, the line segment is displayed on the screen.

The method of animation presented here is unusual in that it completely separates the two processes. The computation of the coordinates of every line segment for every screen image is done by a BASIC program which writes the coordinates to disk as a binary (non-ASCII) file. This file of line segment coordinates is then input to a machine language program which displays the screens in rapid succession to produce the animation.

To begin, type in and save Program 1. Before you run this program, make sure you have a disk in the active drive with at least 60,000 bytes of available space. Now run Program 1; it creates a disk file named ANIMATE.OBJ containing the machine language animation routine. The DOS LINK utility must then be used to generate an executable version of this file. To do this, first exit DOS by typing SYSTEM and pressing Enter. Place a DOS system disk containing the file LINK .EXE in the active drive (check the master disk that came with your copy of DOS), type LINK, then press Enter. When you are prompted for the object modules, remove the DOS system disk and replace it with the disk containing ANIMATE .OBJ. At this point you should type **ANIMATE,,NUL,NUL** and press Enter. After a minute or so the DOS prompt will reappear. Your disk now contains a new file named ANIMATE.EXE, the usable version of the machine language program that creates animated displays from the files produced by Programs 2 or 3.

A Rotating Demo

Now you are ready to type in and save Program 2 (this program can be saved on any disk). When you run the program, you will be prompted for an output filename. Enter any legal filename. Program 2 creates images of the word LOVE rotating in three-dimensions. After you press Enter, the program begins computing the line segment coordinates for each screen and writing them to the specified disk file. The display will show which screen is currently being computed.

Program 2 computes 71 screens. Do not remove the disk from the drive until you see the message that the file is complete.

When the BASIC Ok prompt reappears, type SYSTEM and press Enter to exit to DOS. Put the disk containing ANIMATE.EXE in the active drive, then type ANIMATE and press Enter. When you are asked for the name of the input file, put the disk containing the file created by Program 2 in the active drive and enter the name you specified for that file. The disk drive light will go on for a few seconds, and then the animated image should appear on the screen. Press the Q key to terminate the display.

Once you have used Program 2 to create the animation data file, you won't need it again. However, before you delete it, notice that lines 430–520 also occur in Program 3. In fact, you'll find these lines in every program that you write that produces data files for the ANIMATE program. To save yourself a lot of typing, load Program 2 and delete all lines except 430–520; save the shortened program with a name you'll remember—you will probably use it as a template program many times.

To enter Program 3, first load the file containing lines 430–520 of Program 2. Then type in the other lines listed as Program 3 and save the file. At this point you should follow the same procedure as for Program 2. Program 3 computes 95 screens. The computation for each screen takes longer than those in Program 2 because of computations to remove hidden lines from the display. Now run the animate program using this data file as input. You will see a rotating cube repeatedly coming toward and going away from you (see photos).

Make Your Own Art

Writing your own programs with ANIMATE is not difficult. Just follow these steps:

1. Load the template file containing the lines 430–520.

2. All DIM statements and initialization of variables should be performed prior to line 430. If there is not enough room in the program to do this, you can GOSUB to a routine located further down in the program. DATA statements, of course, can be placed anywhere in the program.

3. The variable NUMSCNS should be assigned a value equal to the number of screens to be displayed. This assignment must also be done prior to line 430.

4. The subroutine that does the computation for each screen must begin at line 1000. For each line segment, the program must compute the segment endpoint coordinates (the variables X1, Y1, X2, and Y2) and execute a GOSUB 500.

The ANIMATE program can handle up to 4000 line segments. This means that the number of screens times the number of line segments per screen cannot exceed 4000.

Programs 2 and 3 both produced 3-D images, but this doesn't mean that you need to know 3-D geometry to create impressive displays. Two-dimensional animation, when it's fast and smooth, can be truly spectacular as well.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

Program 1: ANIMATE.OBJ File Maker

```
KN 10 T=0:DPEN "ANIMATE.OBJ" FOR
       DUTPUT AS 1
KL 20 FOR J=1 TO 1076:READ A$:N=
      VAL("&H"+A$)
FA 30 T=T+N:PRINT#1,CHR$(N);:NEX
      T:CLDSE 1
PL 40 IF T=84992! THEN PRINT"FIL
      E SUCCESSFULLY CREATED!":E
      ND
LM 50 PRINT CHR$(7);"***** ERRDR
       IN DATA STATEMENTS *****"
      :END
BE 100 DATA 80,03,00,01,41,38,96
       ,11,00,00
OD 110 DATA 04,43,53,45,47,04,44
       ,53,45,47
BI 120 DATA 04,53,53,45,47,D6,9B
       ,07,00,60
OM 130 DATA E1,01,02,01,01,1B,98
       ,07,00,60
NB 140 DATA 9D,8F,03,01,01,A0,98
       ,07,00,74
LM 150 DATA 80,00,04,01,01,67,A0
       ,0C,00,02
AA 160 DATA 00,00,80,40,20,10,0B
       ,04,02,01
JL 170 DATA 53,A2,0F,00,02,08,00
       ,00,40,01
EF 180 DATA 00,01,00,00,00,02,00
       ,00,01,A2
BO 190 DATA 0F,00,02,08,B0,A0,1F
       ,01,00,01
PG 200 DATA 00,00,00,02,00,00,02
       ,A0,1A,00
GC 210 DATA 02,4B,BF,00,00,00,00
       ,00,00,00
BE 220 DATA 00,00,00,00,00,00,00
       ,00,00,00
JH 230 DATA 00,00,00,14,00,29,A2
       ,0E,00,02
MH 240 DATA 5E,8F,14,00,01,00,01
       ,00,00,00
OC 250 DATA 01,20,FA,A0,2F,00,02
       ,72,BF,00
NK 260 DATA 00,45,6E,74,65,72,20
       ,69,6E,70
DF 270 DATA 75,74,20,66,69,6C,65
       ,20,6E,61
MM 280 DATA 6D,65,3A,20,24,0A,0D
       ,46,69,6C
DP 290 DATA 65,20,6E,6F,74,20,66
       ,6F,75,6E
II 300 DATA 64,24,49,A0,01,01,01
       ,00,00,1E
MI 310 DATA 33,C0,50,88,00,00,8E
       ,D8,B8,00
DB 320 DATA 06,87,07,89,00,00,BA
       ,4F,18,CD
DJ 330 DATA 10,33,D2,87,00,84,02
       ,CD,10,8D
IO 340 DATA 16,00,00,84,09,CD,21
       ,8D,16,00
OL 350 DATA 00,84,0A,CD,21,87,00
       ,8A,1E,00
GC 360 DATA 00,C6,87,00,00,00,8D
       ,16,00,00
NK 370 DATA 80,00,84,3D,CD,21,73
       ,09,8D,16
IA 380 DATA 00,00,84,09,CD,21,C8
       ,A3,00,00
IN 390 DATA 88,1E,00,00,8D,16,00
       ,00,52,89
PG 400 DATA 80,00,84,3F,CD,21,5A
       ,81,C2,80
OB 410 DATA 00,3D,00,00,75,EE,88
       ,06,00,CD
PI 420 DATA 10,E8,00,00,8D,1E,00
       ,00,88,07
FJ 430 DATA 3D,9D,FF,74,2F,3D,19
       ,FC,75,05
FI 440 DATA E8,00,00,E8,E8,A3,00
       ,00,83,C3
JK 450 DATA 02,88,07,A3,00,00,83
       ,C3,02,88
OO 460 DATA 07,A3,00,00,83,C3,02
       ,88,07,A3
BK 470 DATA 00,00,83,C3,02,53,E8
       ,00,00,58
GC 480 DATA E8,CA,E8,00,00,84,06
       ,82,FF,CD
NE 490 DATA 21,3C,71,74,04,3C,51
       ,75,18,32
BB 500 DATA FF,88,00,06,33,C9,8A
       ,4F,18,CD
LO 510 DATA 10,88,00,02,33,D8,33
       ,D2,CD,10
NL 520 DATA 88,02,00,CD,10,C8,E8
       ,00,00,83
HP 530 DATA C3,02,EB,96,1E,06,8C
       ,D8,8E,C0
PB 540 DATA BD,3E,00,00,88,00,88
       ,8E,D8,33
DN 550 DATA F6,89,A0,1F,FC,F3,A5
       ,07,1F,C3
IB 560 DATA 06,88,45,9C,94,00,C8
       ,05,00,02
GB 570 DATA 02,9D,8F,C4,20,00,02
       ,02,74,8F
EH 580 DATA C4,28,00,02,02,5C,8F
       ,C4,32,00
FL 590 DATA 02,02,5D,BF,C4,36,00
       ,02,02,5E
EI 600 DATA 8F,C4,3B,00,02,02,5E
       ,8F,C4,47
LB 610 DATA 00,02,02,8C,8F,C4,4F
       ,00,02,02
OF 620 DATA 72,8F,C4,53,00,02,02
       ,72,8F,C4
```

```
BD 630 DATA 57,00,02,02,0B,00,B4
       ,71,00,01
KD 640 DATA 01,E1,00,C4,75,00,02
       ,02,0B,00
NB 650 DATA B4,B4,00,01,01,0F,01
       ,C4,B9,00
MB 660 DATA 02,02,4B,BF,C4,91,00
       ,02,02,4A
LE 670 DATA BF,C4,99,00,02,02,4C
       ,BF,C4,A1
BF 680 DATA 00,02,02,4E,BF,B4,AB
       ,00,01,01
PK 690 DATA 22,01,B4,AE,00,01,01
       ,F8,00,B4
FA 700 DATA DA,00,01,01,0F,01,C4
       ,E9,00,02
LD 710 DATA 02,0B,B0,1B,A0,EB,00
       ,01,FD,00
DK 720 DATA 00,BB,BE,C0,B9,A0,1F
       ,33,FF,BD
DB 730 DATA 36,00,00,FC,F3,A5,07
       ,C3,06,BC
PB 740 DATA DB,BE,C0,B9,A0,1F,BD
       ,3E,00,00
JI 750 DATA 33,C0,FC,F3,AB,07,C3
       ,06,8C,DB
JC 760 DATA 8E,C0,8E,01,00,8F,01
       ,00,8B,16
JB 770 DATA 00,00,2B,16,00,00,7D
       ,04,F7,DF
MA 780 DATA F7,DA,89,3E,00,00,88
       ,0E,00,00
FP 790 DATA 28,0E,00,00,7D,04,F7
       ,DE,F7,D9
AE B00 DATA 89,36,00,00,3B,CA,7D
       ,08,BE,00
BL 810 DATA 00,87,CA,E8,04,90,BF
       ,00,00,B9
DJ 820 DATA 36,00,00,89,3E,00,00
       ,BB,C2,01
LJ 830 DATA E0,A3,00,00,2B,C1,B8
       ,08,28,C1
DC 840 OATA A3,00,00,8B,36,00,00
       ,88,3E,00
DJ 850 DATA 00,41,56,53,B8,C7,8A
       ,E0,25,FE
BH 860 DATA 01,D1,E0,D1,E0,D1,E0
       ,88,D8,80
ME 870 DATA E7,07,D1,E0,01,E0,03
       ,D8,8D,06
BK 880 OATA 00,00,03,08,8B,C6,D1
       ,FB,01,FB
DB 890 DATA D1,F8,03,D8,81,E6,07
       ,00,8A,84
QP 900 DATA 00,00,26,0A,07,26,88
       ,07,5B,5E
LI 910 DATA 83,F8,00,70,11,03,36
       ,00,00,03
KC 920 DATA 3E,00,00,03,1E,00,00
       ,E2,B3,EB
MM 930 DATA 0F,90,03,36,00,00,03
       ,3E,00,00
EJ 940 DATA 03,1E,00,00,E2,A2,07
       ,C3,B2,9C
JP 950 DATA 99,00,C4,08,00,02,02
       ,08,80,C4
EJ 960 DATA 1C,00,02,02,08,B0,C4
       ,32,00,02
DD 970 DATA 02,4E,8F,C4,36,00,02
       ,02,4A,8F
KD 980 DATA C4,40,00,02,02,52,8F
       ,C4,44,00
HI 990 OATA 02,02,4C,BF,C4,48,00
       ,02,02,48
DP 1000 DATA 8F,C4,52,00,02,02,5
        0,BF,C4,65
MM 1010 DATA 00,02,02,54,BF,C4,6
        9,00,02,02
HM 1020 DATA 56,BF,C4,70,00,02,0
        2,58,BF,C4
DH 1030 DATA 79,00,02,02,5A,BF,C
        4,7D,00,02
CL 1040 DATA 02,48,8F,C4,81,00,0
        2,02,4A,BF
BD 1050 DATA C4,A0,00,02,02,0B,B
        0,C4,B4,04
MP 1060 DATA 02,02,C4,C5,00,02,0
        2,54,BF,C4
CP 1070 DATA C9,00,02,02,56,BF,C
        4,C0,00,02
IA 10B0 DATA 02,5B,BF,C4,06,00,0
        2,02,50,BF
PM 1090 OATA C4,DA,00,02,02,52,B
        F,C4,OE,00
PB 1100 DATA 02,02,5A,BF,3B,90,0
        E,00,00,01
MH 1110 DATA 07,41,52,52,59,53,4
        3,4E,FB,00
BK 1120 OATA 00,3D,90,0C,00,00,0
        1,05,45,52
IN 1130 OATA 41,53,45,0F,01,00,D
        E,90,0E,00
JD 1140 OATA 00,01,07,4D,45,4D,4
        C,49,4E,45
EK 1150 OATA 22,01,00,30,90,0E,0
        0,00,01,07
CH 1160 OATA 53,43,4E,41,52,52,5
        9,E1,00,00
HE 1170 OATA 57,BA,02,00,00,74
```

Program 2: LOVE File Maker

```
DJ 10 OIM BX(11),BY(11),EX(11),E
      Y(11)
PH 20 FDR N=0 TO 11:READ BX(N),B
      Y(N),EX(N),EY(N):NEXT
DC 30 DATA -22,3,-22,-3,-22,-3,-
      14,-3
DA 40 DATA -10,3,-10,-3,-10,-3,-
      2,-3
EL 50 DATA -2,-3,-2,3,-2,3,-10,3
JB 60 DATA 2,3,6,-3,6,-3,10,3
FB 70 DATA 22,3,14,3,14,3,14,-3
DJ 80 DATA 14,-3,22,-3,20,0,14,0
BA 90 CX=320:CY=100:A=6.2831853#
CF 100 NUMSCNS=71
JN 430 INPUT"DUTPUT FILE NAME";F
       $:OPEN F$ FDR OUTPUT AS 1
EE 440 PRINT"CDMPUTING SCREEN NU
       MBER: ";
MD 450 FOR SCRN=1 TD NUMSCNS:PRI
       NT SCRN;
MH 460 GDSUB 1000
IE 470 PRINT#1,CHR$(157);CHR$(25
       5);:NEXT SCRN
FD 480 PRINT#1,CHR$(25);CHR$(252
       ):CLOSE 1:PRINT
BJ 490 PRINT"ANIMATIDN OATA FILE
        ";CHR$(34);F$;CHR$(34);"
        IS CDMPLETE":END
CL 500 PRINT#1,CHR$(INT(X1) ANO
       255);CHR$(INT(X1/256));CH
       R$(INT(Y1));CHR$(0);
KA 510 PRINT#1,CHR$(INT(X2) AND
       255);CHR$(INT(X2/256));CH
       R$(INT(Y2));CHR$(0);
MD 520 RETURN
HF 1000 FDR N=0 TD 11
KL 1010 ZE=-BX(N)*S1N(A)+30
PD 1020 X1=100*BX(N)*CDS(A)/ZE+C
        X:Y1=-100*BY(N)/ZE+CY
MP 1030 ZE=-EX(N)*SIN(A)+30
ML 1040 X2=100*EX(N)*CDS(A)/ZE+C
        X:Y2=-100*EY(N)/ZE+CY
II 1050 GDSUB 500
DI 1060 NEXT N:A=A-8.726646E-02
JA 1070 RETURN
```

Program 3: CUBE File Maker

```
BF 1 ' PRDGRAM 3
DH 2 '
MN 10 DIM V(B,3),SV(8,2),S(6,5),
      N(6,3),E(12,3)
FE 20 FOR I=1 TD B:FDR J=1 TD 3:
      READ V(I,J):NEXT J,I
AG 30 FDR I=1 TD 6:FDR J=1 TD 5:
      REAO S(I,J):NEXT J,I
KN 40 DATA 40,40,-40,40,40,40,40
      ,-40,40,40,-40,-40
PJ 50 DATA -40,-40,-40,-40,-40,4
      0,-40,40,40,-40,40,-40
LE 60 DATA 1,2,3,4,1,1,B,7,2,1,B
      ,5,6,7,B
ME 70 DATA 5,4,3,6,5,2,7,6,3,2,4
      ,5,B,1,4
KP 90 CX=320:CY=100:TH=.2:PH=.B:
      PPO=2000:DIST=20000
MH 100 NUMSCNS=95
JN 430 INPUT"OUTPUT FILE NAME";F
       $:OPEN F$ FOR OUTPUT AS 1
EE 440 PRINT"CDMPUTING SCREEN NU
       MBER: ";
MB 450 FDR SCRN=1 TO NUMSCNS:PRI
       NT SCRN;
MH 460 GOSUB 1000
IE 470 PRINT#1,CHR$(157);CHR$(25
       5);:NEXT SCRN
FD 480 PRINT#1,CHR$(25);CHR$(252
       ):CLDSE 1:PRINT
BJ 490 PRINT"ANIMATIDN DATA FILE
        ";CHR$(34);F$;CHR$(34);"
        IS COMPLETE":ENO
CL 500 PRINT#1,CHR$(INT(X1) AND
       255);CHR$(INT(X1/256));CH
       R$(INT(Y1));CHR$(0);
KA 510 PRINT#1,CHR$(INT(X2) AND
       255);CHR$(INT(X2/256));CH
       R$(INT(Y2));CHR$(0);
MD 520 RETURN
KC 1000 S1=SIN(TH):C1=CDS(TH):S2
        =SIN(PH):C2=COS(PH)
PP 1010 FOR I=1 TD 8:X=V(I,1):Y=
        V(I,2):Z=V(I,3):SX=-X*S1
        +Y*C1
FL 1020 SY=-X*C1*C2-Y*S1*C2+Z*S2
        :SZ=-X*S2*C1-Y*B2*S1-Z*C
        2+OIST
IB 1030 SV(I,1)=PPD*(2.67*SX/SZ)
        +CX:SV(I,2)=-PPD*(SY/SZ)
        +CY:NEXT
ID 1040 FOR I=1 TD 6:F=S(I,1):G=
        S(I,2):H=S(I,3):U1=V(G,1
        )-V(F,1):U2=V(G,2)-V(F,2
        )
FL 1050 U3=V(G,3)-V(F,3):V1=V(H,
        1)-V(F,1):V2=V(H,2)-V(F,
        2):V3=V(H,3)-V(F,3)
BJ 1060 N(I,1)=U2*V3-V2*U3:N(I,2
        )=U3*V1-V3*U1:N(I,3)=U1*
        V2-V1*U2:NEXT
MI 1070 XE=DIST*S2*C1:YE=O1ST*S2
        *S1:ZE=OIST*C2:M=1
JD 1080 FOR I=1 TD 6:E2=S(I,1):W
        X=XE-V(E2,1):WY=YE-V(E2,
        2):WZ=ZE-V(E2,3)
MH 1090 IF (N(I,1)*WX+N(I,2)*WY+
        N(I,3)*WZ)<=0 THEN 1140
DI 1100 E1=S(I,1):FDR J=2 TD 5:E
         2=8(I,J):FDR K=1 TO M
KH 1110 IF E(K,1)=E2 AND E(K,2)=
        E1 THEN E(K,3)=2:GOTO 11
        30
EP 1120 NEXT K:E(M,1)=E1:E(M,2)=
        E2:E(M,3)=1:M=M+1
BH 1130 E1=E2:NEXT J
CB 1140 NEXT I:FDR I=1 TD 12:IF
        E(I,3)=0 THEN 1160
CH 1150 J=E(I,1):K=E(I,2):X1=SV(
        J,1):Y1=SV(J,2):X2=SV(K,
        1):Y2=8V(K,2):GOSUB 500
PC 1160 NEXT:TH=TH+6.544985E-02:
        PH=PH+6.544985E-02:IF SC
        RN<48 THEN PPO=PPO+583.3
        :RETURN
IA 1170 PPD=PPD-583.3:RETURN
```

©

COMPUTE!'s *All New* Apple Applications Special

COMPUTE!'s latest *Apple Applications Special* features in-depth articles and interviews, all the inside news about Apple, clearly written tutorials, software buyer's guides, new product information, and valuable ready-to-type-in programs for all Apple users.

Off The Beaten Software Path:
Practical, Unusual, And Wonderful Computer Programs You Can Buy

COMPUTE!'s Apple APPLICATIONS SPECIAL

$3.95
Vol 2, No. 1

Apple's Complete, Official User Group List. This Issue

'86 Apple: An Interview With John Sculley

Business Software Buyer's Guide

The Expanding Macintosh
The Debut Of The Macintosh Plus

Apple Rules The Schools
The Most Popular Computer In Education

Educational Software Buyer's Guide

Reviews
Balance of Power
Fantavision
Managing Your Money

Apple Applications, Part Of The COMPUTE! Applications Series Published Four Times A Year

Ready To Type In And Use Programs:

MouseCursor
Redesign Your Macintosh Cursor With This Easy-To-Master Program

Windows
Create And Define Macintosh-like Windows On Your Apple II

Lexitron
A Challenging Word Puzzle

APPLE II DISK Available, see page 35

Apple owners find these special Apple issues the most understandable, complete, and valuable resources available today.

PLUS

All the programs in *COMPUTE!'s Apple Applications Special* are also available on a timesaving disk, ready to run on your Apple II, II+, IIe, and IIc. The *Disk* costs only $12.95 (plus $2.00 shipping and handling) and gives you immediate access to all the great programs in this special issue.

Look for the Fall/Winter 1986 issue of *COMPUTE!'s Apple Applications Special* on sale where you buy other COMPUTE! publications, or order directly from COMPUTE!. This special issue goes on sale October 7, 1986.

Send in the attached order card or call toll free 800-346-6767 (in NY call 212-887-8525).

Features

The Latest At Apple

An inside look at the newest products from Apple. Impressive hardware, sophisticated software.

Interview: William Mensch

A fascinating talk with the designer of the chips inside the millions of Apple computers. What we can expect in the coming years.

Business Applications

Mastering The Macintosh

Power software for the Macintosh takes Apple's innovative computer to new heights. Complete tutorial and application ideas.

New On The Wire

Telecommunications keeps exploding—witness the new information services, new software, and powerhouse 2400 baud modems. Our telecommunications specialists look at all this and more.

At School

Experts At The Table

A panel of well-known software developers, educators, and computerists explain where educational computing has been, where it's going, and why.

The Story Machine

Children of all ages can write, read, and print stories up to 20 pages long with this easy-to-use word processor for all Apple II computers.

Solarpix

Educational software that entertains, this tour of the solar system is a dazzling display of the Apple's graphics power.

Hi-Res Poster Printer

Print hi-res drawings in poster size for banners, signs, and business presentations. Two sizes and a simple-to-use menu make it a snap.

At Home

Your Graphics Primer

Tips, techniques, and more show how to create impressive graphics on the Apple II.

Power Sketch

This hi-res sketching program for the Apple II puts paint, draw, fill, air brush, and other tools in every artist's hands. Save and load screens to disk.

High Rise

This unique game of elevators and an out-of-control robot is written entirely in machine language. Easily one of the best Apple arcade-style games we've ever published.

DOS Adjust

Customize DOS 3.3 with this comprehensive program that's easy to use: no programming knowledge required.

Apple User Groups

A complete list of all Apple user groups.

And much more

Including reviews, buyer's guides, new products, and useful programs.

COMPUTE!™ Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications.

PowerKey For Apple

Patrick Parrish, Programming Supervisor

This valuable utility puts 52 customized strings or keywords instantly at your fingertips. You can even create several sets of custom commands for use with different applications. For all Apple II series computers with DOS 3.3 or ProDOS.

Using an Apple II computer usually involves a considerable amount of typing, and most of us—good typists or not—would be happy to automate the process of communicating with our machine. Have you ever wished you could just strike one key and produce a directory, run a program, or perform some common task?

"PowerKey" provides a selection of 52 different one-touch keywords which you can customize to your own liking. It lets you access up to 52 keywords or other strings of your own by pressing either the Open Apple or Solid Apple key (or paddle buttons for those who have an Apple II+, which lacks these keys) along with one of the letter keys (A–Z). Although the program relies on a short machine language routine, you can use it without understanding machine language at all.

Entering The Program

This utility is written in three parts. Program 1, POWERKEY.CUSTOM, is a BASIC program that lets you create and save tables of your custom strings or keywords to disk. Program 2, POWERKEY.LOADER, is a BASIC loader which POKEs the machine language driver routine into memory and saves a copy of this code to disk in the form of a binary file. (Since Program 2 uses the name POWERKEY.BINARY for the file it creates, you *must not* use that name for Program 2 itself. If you do, you'll get a FILE TYPE MISMATCH error when Program 2 is run.) Program 3, POWERKEY.SYSTEM, is a short BASIC program which loads both the keyword table and the driver routine, and then activates PowerKey. Before going any further, carefully type in these three programs and save a copy of each to disk.

Creating Customized Keys

After entering Programs 1–3, load and run Program 1, which creates a customized table of keywords and strings. The first prompt asks if you want to load a keyword table from disk. Since this is the first time you've run the program, no tables yet exist, so you should press N for no. In the future, after creating one or more tables, you could also press Y to gain access to a preexisting table. If you press Y, the program displays a directory and asks you to enter the filename of the table to load. If you press RETURN at this prompt without entering a name, PowerKey looks for a default file named TABLE.

If you've specified that no keyword table is to be loaded, Program 1 reads in its 52 default keywords (see lines 910–960). The first 26 keywords can be accessed with the Open Apple key (or the paddle 0 button), and the second 26 keywords by the Solid Apple key (or the paddle 1 button). You can change or rearrange the keywords in the DATA statements if you like, but make sure not to add or delete any keywords. You'll get an OUT OF DATA error if there aren't at least 52 DATA items.

Now PowerKey displays keywords 1–26 on the screen. To the left of each keyword is the letter that will access it. For instance, the keyword AND is represented by *A*. Each keyword or string in the table can be up to 16 characters long, but they can be combined for longer commands. A table can occupy a maximum of 832 bytes and unused characters are signified by dots.

At the bottom of the screen, you are given three options. You can press A to Alter a keyword, the

Solid Apple key (or paddle 1 button) to look at the second 26-keyword set, or W to write the completed table to a disk file. You can switch back and forth between keyword sets by pressing the Solid Apple key (or paddle 1 button) and Open Apple key (or paddle 0 button).

For practice, let's change CATALOG, the current default keyword accessed with Open Apple-C. Select the Open Apple keyword set, then press A and the program prompts you for the letter of the keyword you wish to change. Enter C for the keyword CATALOG. Let's add a carriage return to this keyword so that you'll be able to examine the disk directory from immediate mode with only one keystroke. Type CATALOG followed by a backslash (\), then press RETURN. The backslash always stands for a carriage return character.

The screen should now reflect the change you've made. Notice that the backslash is shown as a control character (CTRL-M is equivalent to RETURN). Other keywords or strings in the table can be altered in the same manner. In fact, if you anticipate repeatedly using a phrase longer than 16 characters in your programming, stretch it out over two or more 16-character strings.

Once the table suits you, press W (for Write) to save it to disk. At this point, the 52 strings in your table are converted to their ASCII equivalents and POKEd into memory at 37376. To distinguish a string from the one that follows, the last character of each string has its high bit set (128 is added to its ASCII value). Before the program saves the table, you are allowed once more to look at the directory on the target disk. After this, a filename for your table is requested. Again, if you strike RETURN, the default filename TABLE is chosen for you. Before the program ends, you are given a chance to put a copy of this file on other disks as well.

Installing The Driver

With the keyword table safely on disk as well as in memory, run Program 2. Line 110 of this program POKEs the PowerKey ML driver routine into memory at 768. This area is safe from BASIC, so PowerKey should not interfere with, or be overwritten by, most programs. Line 130 saves a copy to disk using the filename POWERKEY.BINARY.

PowerKey is now ready to be activated. Type CALL −768 and press RETURN. Then, press Open Apple (or paddle button 0) along with the A key. The keyword AND should appear on the screen. Press RETURN and try another one. Hit Open Apple and C for CATALOG. Immediately, a directory of your disk appears on the screen (recall that we added a carriage return to CATALOG).

Try out some more keywords, using the Solid Apple (paddle button 1) set as well. The computer recognizes your keywords and strings from immediate and program mode as well as from the monitor.

Putting It All Together

Because PowerKey is on your disk as a binary file, it can easily be loaded and run by other programs. In fact, this is just what Program 3 does. It sets HIMEM to protect the reserve space for the keyword table, then asks you to specify the name of the table to be loaded from disk (press RETURN alone at the prompt to load the default file TABLE). The POWERKEY.BINARY machine language file created by Program 2 is loaded into memory, and activated with the appropriate CALL. You can even have PowerKey automatically loaded when you boot your disk if you use DOS 3.3. Simply save Program 3 as the HELLO file on the desired disk.

You can also load PowerKey from immediate mode. With DOS 3.3, type in the following line (substitute the appropriate table filename for TABLE):

```
HIMEM:37376:PRINT CHR$(4)"BLOAD
  TABLE,A37376":PRINT CHR$(4)
  "BLOAD POWERKEY.BINARY":CALL
  768
```

If you are using ProDOS, substitute this line:

```
HIMEM:36352:PRINT CHR$(4)"BLOAD
  TABLE,A37376":PRINT CHR$(4)
  "BLOAD POWERKEY.BINARY":CALL
  768
```

How It Works

PowerKey works basically the same whether you are using DOS 3.3 or ProDOS. In either operating system, it relies on a method known as a *wedge*. The input vector that normally points to the keyboard input subroutine (KEYIN) at $FD1B, is rerouted to point instead to the starting location of our machine language code. Once this is done, the program checks a flag to see whether it is already in the process of printing a keyword. If not, it checks the Open and Solid Apple keys. The routine also responds to paddle button presses, since the Open and Solid Apple keys are read by the same circuitry that reads the buttons.

If one of the special keys is pressed, PowerKey begins printing the one-touch keyword. First, the relative number (0–51) of the desired keyword is determined, a flag is set, and the keyword is located in the table. The first character of the word is then put in the accumulator, the table location is updated, and we return to BASIC. The operating system then prints the character in the accumulator and returns to the program for another character. The next time through the program, another character is loaded into the accumulator since the flag is set. This process continues until the last character of the keyword or phrase is detected (this character has the high bit set). The flag is then set to zero and we're returned to BASIC.

Before all this can happen, however, the program must go through a short initialization routine to determine which operating system is installed. This is done by looking at the starting location for ProDOS's global page ($BF). When ProDOS has been booted, the value in location $BF00 is always 76 (representing the JMP command). If this is the case, then the vector that points to KEYIN (CHIN1 at $BE32–$BE33) is loaded in low-byte/high-byte format with the starting address for our routine, and the program returns to BASIC.

If the value at $BF00 is some other value, then the program assumes we are in DOS 3.3. In this case, the input vectors (KSW for KeySWitch) at $38–$39, which normally point to KEYIN, are loaded in a likewise manner with the starting address of our program. We then jump to a routine at $3EA which updates the input pointers with these new values, reconnects DOS, and returns us to BASIC. Henceforth, with either operating system, our routine gets called so we can print our keywords.

Program 1: Keyword Table Customizer

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
97 1ØØ REM OMNIKEY.CUSTOM
C6 11Ø TEXT :TL = 37376
9A 12Ø HIMEM: TL: IF PEEK (4BB96
       ) = 76 THEN HIMEM: TL - 1
       Ø24: REM TL IS TABLE LOCA
       TION; IF PROOOS, HIMEM IS
        MOVEO OOWN 1K MORE
2B 13Ø FOR I = 768 TO 777: READ
       A: POKE I,A: NEXT : OATA
       1Ø4,16B,1Ø4,166,223,1S4,7
       2,152,72,96: REM ONERR FI
       X
Ø9 14Ø OIM A$(52):F$ = "........
       .........":R$(Ø) = "OPEN-A
       PPLE":R$(1) = "SOLIO-APPL
       E":P = Ø: REM APPLE KEYS
       CORRESPONO TO PAOOLE BUTT
       ONS
D5 15Ø HOME : HTA8 11: VTA8 6: I
       NVERSE : PRINT "KEYWORO C
       USTOMIZER": NORMAL
C1 16Ø VTA8 1Ø: PRINT "WANT TO L
       OAO A TA8LE FROM OISK";:
       GOSUB 79Ø
F9 17Ø IF X < > B9 THEN 2ØØ
55 1BØ GOSU8 83Ø
E7 19Ø W = Ø:V = 13: GOSU8 4BØ:
       GOTO 21Ø
7F 2ØØ FOR I = 1 TO 52: REAO A$(
       I): NEXT
4C 21Ø GOSUB 2BØ
5D 22Ø X = PEEK ( - 16384):Y = P
       EEK ( - 16287):Z = PEEK (
        - 162B6): IF X < = 127 A
       NO Y < = 127 ANO Z < = 12
       7 THEN 22Ø
5A 23Ø POKE - 16368,Ø:X = X - 12
       B: IF X = 87 THEN 57Ø
94 24Ø IF Z > 127 ANO P = Ø THEN
        P = 1: GOTO 21Ø
B6 25Ø IF Y > 127 ANO P = 1 THEN
        P = Ø: GOTO 21Ø
9D 26Ø IF X = 65 THEN 36Ø
1C 27Ø GOTO 22Ø
3Ø 28Ø HOME : VTA8 2: HTA8 11: I
       NVERSE : PRINT R$(P);: NO
       RMAL : PRINT " KEYWOROS:"
       : PRINT
9A 29Ø PRINT :L = 1:H = 13: FOR
       J = 1 TO 22 STEP 21: FOR
       I = L TO H: INVERSE : HTA
       B J: PRINT CHR$ (64 + I);
       : NORMAL : PRINT " ";
C4 3ØØ O$ = A$(I + (P = 1) * 26)
       : FOR Z = 1 TO LEN (O$):X
        = ASC ( MIO$ (O$,Z,1)):
       IF X < 32 THEN INVERSE :
       PRINT CHR$ (X + 64);: NOR
       MAL : GOTO 32Ø
9B 31Ø PRINT CHR$ (X);
5A 32Ø NEXT Z: PRINT MIO$ (F$,1,
       16 - LEN (A$(I + (P = 1)
       * 26))): NEXT :L = 14:H =
        26: VTAB 5: NEXT
FA 33Ø VTAB 2Ø: PRINT "PRESS: ";
       : INVERSE : PRINT "A";: N
       ORMAL : PRINT " TO ";: IN
       VERSE : PRINT "ALTER";: N
       ORMAL : PRINT " A KEYWORO
       ,"
8A 34Ø VTAB 21: HTA8 B: INVERSE
       : PRINT R$(P = Ø);: NORMA
       L : PRINT " FOR ";: INVER
       SE : PRINT R$(P = Ø);: NO
       RMAL : PRINT " SET,": HTA
       B B: INVERSE : PRINT "W";
       : NORMAL : PRINT " TO ";:
        INVERSE : PRINT "WRITE";
       : NORMAL : PRINT " TABLE
       TO OISK."
6B 35Ø REM INPUT KEYWORO
91 36Ø VTA8 2Ø: HTAB 2B: PRINT "
        ": HTA8 8: PRINT "

        ": HTAB B: PRINT "
                         "
2B 37Ø VTAB 2Ø: PRINT "ENTER KEY
        (A-Z) TO CHANGE ";: INPU
       T L$:L = ASC (L$) - 64: I
       F L < Ø OR L > 26 OR LEN
       (L$) > 1 THEN 37Ø
71 3BØ VTAB 22: PRINT "NEW KEYWO
       RO FOR ";: INVERSE : PRIN
       T L$;: NORMAL : PRINT " ?
       ";: PRINT F$
BØ 39Ø VTAB 24: PRINT "('\' WILL
        EMBEO A CARRIAGE RETURN)
       ";: HTAB 2Ø: VTAB 22:C =
       Ø:O$ = ""
C7 4ØØ X = PEEK ( - 163B4): IF X
        < = 127 THEN 4ØØ
5B 41Ø POKE - 1636B,Ø:X = X - 12
       B: IF X = 13 THEN 46Ø
BØ 42Ø IF X = 92 THEN X = 13
Ø9 43Ø C = C + 1:O$ = O$ + CHR$
       (X): IF X < 32 THEN INVER
       SE : PRINT CHR$ (X + 64);
       : NORMAL : GOTO 45Ø
9F 44Ø PRINT CHR$ (X);
D7 45Ø IF C < 16 THEN 4ØØ
BB 46Ø A$(L + (P = 1) * 26) = O$
       : FOR I = 1 TO 4ØØ: NEXT
       : GOTO 21Ø
92 47Ø REM LOAO TABLE
C2 48Ø ONERR GOTO 75Ø
47 49Ø HOME : HTAB 6: VTAB 1Ø: G
       OSUB 69Ø: PRINT : PRINT C
       HR$ (4)"BLOAO "FL$: POKE
       216,Ø
57 5ØØ VTAB 16: HTAB 1Ø: PRINT "
       REAOING TABLE..."
74 51Ø C = Ø: FOR I = 1 TO 52:EF
        = Ø
E9 52Ø A = PEEK (TL + C): IF A >
        127 THEN A = A - 128:EF
       = 1
3E 53Ø A$(I) = A$(I) + CHR$ (A):
       C = C + 1: IF EF THEN NEX
       T I
37 54Ø IF I < 53 THEN 52Ø
1F 55Ø RETURN
63 56Ø REM SAVE TA8LE
EF 57Ø HOME : VTAB 7: HTAB 9: NO
       RMAL : PRINT "...SETTING
       UP TABLE"
2F 5BØ C = Ø:A = Ø: FOR I = 1 TO
        52:C = C + A:A = LEN (A$
       (I)): FOR J = 1 TO A - 1
2E 59Ø G = ASC ( MIO$ (A$(I),J,1
       )):: IF G = 92 THEN G = 1
       3
EA 6ØØ POKE TL + C + J - 1,G: NE
       XT J
6D 61Ø 8 = ASC ( RIGHT$ (A$(I),1
       )) + 12B: IF B = 22Ø THEN
        B = 141
19 62Ø POKE TL + C + A - 1,B: NE
       XT I
54 63Ø VTAB 1Ø: HTAB 6: PRINT "R
       EAOY TO SAVE TABLE TO OIS
       K.": GOSUB B3Ø:W = 1: HOM
       E
BC 64Ø ONERR GOTO 75Ø
B1 65Ø VTAB 1Ø: HTAB 6:V = 13: G
       OSUB 69Ø: PRINT : PRINT C
       HR$ (4)"BSAVE "FL$",A" ST
       R$ (TL)",LB32": POKE 216,
       Ø
55 66Ø VTAB 16: HTAB 6: PRINT "A
       NOTHER COPY";: GOSUB 79Ø
F8 67Ø IF X = B9 THEN HOME : GOT
       O 64Ø
9E 6BØ ENO
5B 69Ø PRINT "TABLE FILENAME: ";
       : INPUT FL$: IF FL$ = ""
       THEN FL$ = "TABLE"
CA 7ØØ VTAB V: PRINT "PUT PROPER
        OISK IN ORIVE & HIT <RET
       URN>.";
ØE 71Ø X = PEEK ( - 163B4): IF X
        < = 127 THEN 71Ø
2E 72Ø POKE - 1636B,Ø:X = X - 12
       B: IF X < > 13 THEN 71Ø
1D 73Ø RETURN
C7 74Ø REM OISK ERROR ROUTINE
9F 75Ø PRINT : HTAB B: PRINT "OI
       SK ERROR #" STR$ ( PEEK (
       222))"."
FD 76Ø CALL 76B: VTA8 18: VTA8 2
       Ø: HTAB B: PRINT "TYPE 'C
       ' TO CONTINUE";: GET S$:
       IF W = Ø THEN 4BØ
42 77Ø IF W = 1 THEN HOME : GOTO
        64Ø
18 78Ø HOME :V = 15: GOTO B4Ø
DD 79Ø PRINT " (";: INVERSE : PR
       INT "Y";: NORMAL : PRINT
       "/";: INVERSE : PRINT "N"
       ;: NORMAL :.PRINT ")?"
CC BØØ X = PEEK ( - 163B4): IF X
        < = 127 THEN BØØ
25 B1Ø GET S$: POKE - 1636B,Ø:X
       = X - 12B: IF X < > 7B AN
       O X < > B9 THEN BØØ
1C 82Ø RETURN
11 B3Ø V = 16: VTAB 13: PRINT "N
       EEO A LOOK AT THE CATALOG
        FIRST";: GOSUB 79Ø: IF X
        = 7B THEN RETURN
22 B4Ø W = 2: ONERR GOTO 75Ø
4F B5Ø GOSUB 7ØØ
D6 B6Ø POKE 34,Ø: HOME : HTAB 12
       : PRINT "OISK CATALOG:":
       HTAB B: PRINT "----------
       ----------"
DE B7Ø POKE 34,2: PRINT : PRINT
       CHR$ (4)"CATALOG": POKE 2
       16,Ø
76 BBØ HTAB B: PRINT : PRINT "CA
       TALOG ANOTHER OISK";: GOS
       UB 79Ø: IF X = B9 THEN V
       = 23: GOTO B5Ø
33 B9Ø PRINT : PRINT "PRESS RETU
       RN TO CONTINUE": GOSU8 71
       Ø
43 9ØØ POKE 34,Ø: RETURN
C5 91Ø REM PAOOLE Ø OR OPEN APPL
       E KEY WOROS
9Ø 92Ø OATA ANO,BLOAO ,CATALOG,O
       ATA ,ENO,FOR,GOSUB,HOME,I
       NPUT,GET,REAO,LOAO ,MIO$(
A3 93Ø OATA NEXT,OR,PRINT,STOP,R
       UN ,SAVE ,THEN,TEXT,VTAB,
       WRITE,PEEK,REM,CONT
26 94Ø REM PAOOLE 1 OR CLOSEO AP
       PLE KEY WOROS
Ø2 95Ø OATA ASC(,BRUN ,CLOSE,OEL
       ,OIM,FLASH,GOTO,HTAB,INVE
       RSE,RESTORE,NDRMAL,LIST
FA 96Ø DATA LEFT$(,NEW,DPEN,PDKE
       ,RIGHT$(,RETURN,STR$(,STE
       P,TA8(,VERIFY,INT(,CALL,L
       EN(,CLEAR
```

Program 2:PowerKey Binary File Creator

```
A4 1ØØ REM DMNIKEY.LDADER
D3 11Ø FDR I = 768 TD 939: READ
       A: POKE I,A:X = X + A: NE
       XT
A2 12Ø IF X < > 18Ø1Ø THEN PRINT
        "ERRDR IN DATA STATEMENT
       S.": STDP
E2 13Ø PRINT CHR$ (4)"8SAVE DMNI
       KEY.8INARY,A768,L172"
81 14Ø DATA 162,146,134,7,16Ø,Ø,
       132,6
A4 15Ø DATA 162,33,16Ø,3,173,Ø,1
       91,2Ø1
Ø6 16Ø DATA 76,2Ø8,7,142,5Ø,19Ø,
       14Ø,51
1F 17Ø DATA 19Ø,96,134,56,132,57
       ,76,234
1A 18Ø DATA 3,44,169,3,48,94,32,
       27
28 19Ø DATA 253,72,32,74,255,173
       ,97,192
A3 2ØØ DATA 16,7,169,Ø,141,17Ø,3
       ,24Ø
FA 21Ø DATA 1Ø,173,98,192,16,65,
       169,26
F7 22Ø DATA 141,17Ø,3,1Ø4,56,233
       ,193,48
C1 23Ø DATA 55,2Ø1,26,176,51,24,
       1Ø9,17Ø
4A 24Ø DATA 3,141,17Ø,3,169,255,
       141,169
F9 25Ø DATA 3,173,17Ø,3,24Ø,38,1
       6Ø,Ø
F4 26Ø DATA 162,Ø,23Ø,6,2Ø8,2,23
       Ø,7
91 27Ø DATA 177,6,48,2,16,244,23
       2,236
1D 28Ø DATA 17Ø,3,2Ø8,238,32,63,
       255,23Ø
D8 29Ø DATA 6,2Ø8,9,23Ø,7,2Ø8,5,
       1Ø4
ØF 3ØØ DATA 32,63,255,96,16Ø,Ø,1
       77,6
6C 31Ø DATA 141,171,3,23Ø,6,2Ø8,
       2,23Ø
88 32Ø DATA 7,173,171,3,48,4,24,
       1Ø5
57 33Ø DATA 128,96,169,Ø,141,169
       ,3,133
23 34Ø DATA 6,169,146,133,7,173,
       171,3
11 35Ø DATA 96,Ø,Ø,Ø
```

Program 3: PowerKey Loader

```
75 1ØØ REM DMNIKEY.SYSTEM
C6 11Ø TEXT :TL = 37376
1A 12Ø HIMEM: TL: IF PEEK (48896
       ) = 76 THEN HIMEM: TL - 1
       Ø24
AA 13Ø HOME : PRINT "ENTER KEYWO
       RD TABLE NAME";: INPUT N$
       : IF N$ = "" THEN N$ = "T
       ABLE"
A5 14Ø PRINT CHR$ (4)"BLOAD "N$"
       ,A" STR$ (TL)
AØ 15Ø PRINT CHR$ (4)"BLOAD OMNI
       KEY.BINARY"
68 16Ø CALL 768: PRINT : PRINT "
       OMNIKEY IS ACTIVATED.": E
       ND
```

©

Atari 130XE Automated RAM Disk

Stephen J. Rockower

Offering high speed and instantaneous access to programs and files, the Atari 130XE's RAM disk is one of its most attractive features. Now it's even more effective with this utility that moves selected programs and files into the RAM disk automatically whenever you boot the system. Your system will be custom configured on power up. A floppy disk drive and Atari DOS 2.5 are required.

If you own an Atari 130XE, you may have a number of BASIC programs or other files which you like to put on the RAM disk whenever you boot up. Once in the RAM disk, those files are available almost instantly, but it's a tedious process to copy each file to RAM manually. "RAM Disk Loader" for the Atari 130XE automates that chore with a custom AUTORUN.SYS file. When you boot the system, it automatically transfers selected BASIC programs and text files from the default drive (D1:) to the RAM disk (D8:).

Typing Instructions

Here's how to create the RAM Disk Loader. First, boot your computer with DOS 2.5. Go to the DOS menu to select option L; then load SETUP.COM. Use option 2 to create an AUTORUN.SYS file named D1TOD8.SAV. Now go back to BASIC and type in the program.

Note that the DATA statement in line 30 should contain the names of the BASIC programs or text files that you want to transfer to the RAM disk on power-up. When adding these names, include the full name and extender (such as PROG.BAS), but not the drive specifier (don't put D1: at the beginning of the name). Every extender must be exactly three characters long; add extra spaces if necessary to pad the extender to the correct length. The last DATA item in this series must be END which acts as a marker for the end of the list of filenames.

When you type line 40, substitute the name of the program you want to run when the system boots. For example, if you want to run MYPROG.BAS from drive D1:, line 40 should look like this:

```
40 READ F$:IF F$="END" THEN RUN
   "D1:MYPROG.BAS"
```

Note that this program can be one of the programs you just put on the RAM disk (to run such a program, use the D8: drive prefix).

Be very careful when typing lines 290 and 560, which contain tiny machine language routines stored in strings. These strings must be typed correctly, or the computer will probably crash. The REM statements at the end of each line explain exactly which characters to type in the strings. After you finish typing in the program, be sure to save a copy to disk. For the program to work properly, you *must* use the same filename you specified when you created the AUTORUN.SYS file (D1TOD8.SAV). Now you are ready to boot up again. This time, all your programs and data will be on D8.

With only slight modifications, you can use this program to transfer programs from D1: to D2: (rather than to D8:) without having to copy each file manually. This modification allows you to do batch file copies from one drive to another. A second possibility is to eliminate the DATA line altogether and read the filenames from a previously created disk file rather than from DATA. With a statement like INPUT#1,STRING$, you can bring in the name of each file to be transferred. The file could terminate with the name of the next program to run (IF STRING$="END" THEN INPUT#1,STRING$:RUN STRING$).

Program Techniques

The program begins by READing filenames one at a time from the DATA statements in line 30. If the name is not END, the program loops through the directory sectors (361–368) one at a time in search of the file. When the file is found, FLEN holds its length.

The subroutine named GETBYTES determines whether this is a BASIC program or a file containing text or other data. Since the file header for a BASIC program always starts with two zeros, we assume that anything lacking two zeros in the header is not BASIC. The next six pairs of header bytes contain information about the size and location of certain memory pointers. We are interested in the last two bytes, which tell us how many more bytes must be loaded to find the end of the file (DEND). The computation in line 680 adjusts the total number for BASIC program files.

At this point, the program opens an IOCB (Input/Output Control Block) to read the bytes from FROM$ into the string ZZ$. Then ZZ$ is manipulated to allow for text/data (FLEN*125) or a BASIC program (actually held as a string of length BYTES). Before writing the string, we must find the true end of the data. If you think about it, a text file of FLEN characters will have fewer than FLEN*125 bytes. By eliminating the zero bytes—CHR$(0), the heart symbol—we arrive at the true length of the file. This feature, incidentally, makes the program unsuitable for use with machine language files, since ML programs often contain one or more zero bytes.

Once you have this program working, you're likely to find many uses for it. I use it to move a main menu program onto the RAM disk, along with a number of programs and files I use to manage our household accounts. This method takes 20 to 30 percent less time than loading in the same files manually.

Atari 130XE RAM Disk Loader

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
CK 15  POKE 712,148:POKE 559,
       Ø:POKE B,255:POKE 731,
       1
PP 2Ø  OIM A$(12B),ZZ$(125*15
       Ø),F$(15),FROM$(15),RD
       ISK$(15),B$(16),FNAME$
       (16)
DB 25  TRAP 71Ø
KI 3Ø  OATA JUNK.1  ,JUNK.2
       ,D1TOO8.BXE,END
BL 4Ø  READ F$:IF F$="END" TH
       EN RUN "D1:NEXTPROG.SA
       V"
KK 5Ø  FDR SNUM=361 TO 368
PM 6Ø  CLDSE #1:FLEN=Ø
JJ 7Ø  A$=CHR$(Ø):A$(128)=CHR
       $(Ø):A$(2)=A$
FC 9Ø  ORIVE=1:TYPE=82:BUF=AD
       R(A$):GOSUB 26Ø:REM "O
       ISC" ROUTINE
ND 1ØØ  GOSUB 33Ø:REM "OECDOE
        " ROUTINE
JB 11Ø  IF FLEN THEN SNUM=368
BF 12Ø  NEXT SNUM
HI 13Ø  REM
HA 14Ø  FRDM$="D1:":FROM$(LEN
        (FROM$)+1)=F$:ROISK$=
        FRDM$:RDISK$(2,2)="8"
HF 15Ø  GO8UB 6ØØ:REM "GET8YT
        ES"
AB 17Ø  INDEX=8YTES*(8YTES<>Ø
        )+FLEN*125*(8YTEB=Ø)
IJ 18Ø  ZZ$="":ZZ$(1)=CHR$(Ø)
        :ZZ$(INDEX)=CHR$(Ø):Z
        Z$(2)=ZZ$
KD 19Ø  OPEN #2,8,Ø,RDI8K$:DP
        EN #1,4,Ø,FROM$
DC 2ØØ  IDC8=1:TYPE=7:BUF=AOR
        (ZZ$):GDSU8 5ØØ:REM "
        IDC8" FOR READ
NK 21Ø  IF 8YTES>Ø THEN 22Ø
EK 211  IF ZZ$(LEN(ZZ$))=CHR$
        (Ø) THEN ZZ$=ZZ$(1,LE
        N(ZZ$)-1):GOTO 211
FC 212  INOEX=LEN(ZZ$)
MP 22Ø  IOCB=2:TYPE=11:8UF=AO
        R(ZZ$):GOSU8 5ØØ:REM
        "IOC8" FOR WRITE
BE 23Ø  CLOSE #1:CLOSE #2
DD 24Ø  GDTO 4Ø
BD 25Ø  END
IM 26Ø  REM PRDCEOURE "DISC"
KD 27Ø  POKE 779,INT(SNUM/256
        ):POKE 778,SNUM-256*I
        NT(SNUM/256)
JA 28Ø  POKE 769,DR1VE:PDKE 7
        73,INT(8UF/256):POKE
        772,8UF-256*INT(BUF/2
        56):POKE 77Ø,TYPE
NC 29Ø  X=USR(ADR("h S⊏(.}"))
        :REM D.1Ø4,32,83,22B,
        96 or emall h,epace,
        Cap. S, inveree small
         d,ctr1-.
HD 3ØØ  RETURN
HD 31Ø  REM TYPE=82 FOR READ,
         87 FDR WRITE
HJ 32Ø  REM
AL 33Ø  REM PRDCEOURE "DECDDE
        "
CJ 34Ø  FLEN=Ø
AJ 35Ø  FOR A=1 TO 8
NN 36Ø  8$=A$((A-1)*16+1,A*16
        ):IF ASC(8$(1,1))>127
         THEN GDTD 46Ø
BB 37Ø  FLEN=ASC(B$(2))+256*A
        SC(8$(3))
MF 38Ø  FSTART=ASC(8$(4))+256
        *ASC(B$(5))
DB 39Ø  FNAME$=8$(6,13)
KD 394  IF FNAME$(LEN(FNAME$)
        )=" " THEN FNAME$=FNA
        ME$(1,LEN(FNAME$)-1):
        GOTO 394
EP 4ØØ  FNAME$(LEN(FNAME$)+1)
        =".":FNAME$(LEN(FNAME
        $)+1)=B$(14,16)
EJ 41Ø  IF FNAME$=F$ THEN A=B
        :GOTD 47Ø
CK 44Ø  FLEN=Ø
DL 47Ø  NEXT A
HM 48Ø  RETURN
IB 49Ø  REM
ID 5ØØ  REM Procedure "IDC8"
BL 51Ø  REM ASSUMES IOCB ALRE
        AOY OPEN FOR READ OR
        WRITE
LF 52Ø  BLDCK=832+IDCB*16
AD 53Ø  POKE BLDCK+2,TYPE:REM
         READ=7,WRITE=11
LJ 54Ø  POKE BLDCK+5,INT(8UF/
        256):PDKE 8LOCK+4,8UF
        -256*INT(BUF/256)
JD 55Ø  POKE 8LOCK+9,INT(INOE
        X/256):POKE BLOCK+B,I
        NOEX-256*INT(INOEX/25
        6)
KA 56Ø  I=USR(AOR("hhh■LV⊏"),
        IOC8*16):REM h,h,h,in
        veree *, L,V, inveree
         d
FC 57Ø  CLDSE #IDC8
HN 58Ø  RETURN
IC 59Ø  REM
MD 6ØØ  REM PRDCEOURE "GET8YT
        ES"
FN 61Ø  DPEN #1,4,Ø,FRDM$
CF 62Ø  GET #1,I:GET #1,J
DF 63Ø  IF I<>Ø DR J<>Ø THEN
        8YTES=Ø:GDTD 69Ø
CA 64Ø  FDR X=1 TD 6
CI 65Ø  GET #1,I:GET #1,J
DD 66Ø  NEXT X
HK 67Ø  DEND=256*J+I
NH 68Ø  8YTES=OEND-256+14
BJ 69Ø  CLOSE #1
HH 7ØØ  RETURN
HM 71Ø  REM
PD 72Ø  POKE 559,34
IL 73Ø  ? "ERROR ";PEEK(195);
        " AT LINE ";PEEK(186)
        +256*PEEK(187)
```

©

IF-THEN-ELSE For *SpeedCalc*

Anthony Chandler

This tutorial shows you how to get more out of SpeedCalc. *By using clever formulas, you can set up a spreadsheet to perform different computations based on the result of logical IF tests. The techniques apply to any version of* SpeedCalc, COMPUTE!'s *powerful machine language spreadsheet program. (The Commodore 64/128 version of* SpeedCalc *appeared in the January, 1986 issue of* COMPUTE!. *The Apple II and Atari versions were published in February 1986 and March 1986, respectively.)*

SpeedCalc, the versatile spreadsheet program published in the January–March, 1986 issues of COMPUTE!, offers a great variety of built-in functions. It supports all the math operations of BASIC, as well as two new ones (@ave and @sum), but there is no specific mention of how the program can perform conditional operations and make decisions. Here are techniques to make *SpeedCalc* calculate based on the outcome of logical tests modeled after the IF-THEN-ELSE construction in BASIC.

More Than A Glorified Calculator

Many people use a spreadsheet as little more than a glorified calculator: Once a sheet has been set up, you punch a button and the program performs a large number of related calculations. While the result of one calculation frequently serves as input for another, the process doesn't involve anything resembling intelligence on the part of the program. Nevertheless, the *SpeedCalc* spreadsheet program *can* test conditions and take action based on the results. The process works very much like the familiar IF-THEN-ELSE construction in BASIC.

In plain English, a typical IF-THEN-ELSE construction would be translated as, "IF a certain condition is true, THEN do the first task. ELSE if the condition is false, do the second task." A computer can't work with abstract concepts such as truth or falsity, but it's very good at telling the difference between one numeric value and another. When the computer performs an IF test in BASIC, it uses numeric values (usually −1 and 0) to represent true and false, respectively. You can verify this by entering the following statements in BASIC direct mode:

```
A=1:PRINT (A=1)
A=0:PRINT (A=1)
```

In Microsoft BASIC and most other versions, the computer prints −1 and 0, indicating that it uses −1 to represent a true condition and 0 to represent a false condition. The BASICs on Apple II and eight-bit Atari computers use 1 instead of −1 to represent true. To implement IF-THEN-ELSE with a formula in *SpeedCalc*, we can take advantage of the fact that true and false are represented as simple numeric values.

How Many Tests Do You Need?

If you give the matter some thought, you'll discover that only two basic IF tests are needed to cover all possible cases. Here they are:

```
IF A>B THEN (this cell=) C ELSE (this
    cell=) D
IF A<>B THEN (this cell=) C ELSE (this
    cell=) D
```

In these examples the letters A, B, C, and D represent the values contained in particular cells within the spreadsheet. A cell, of course, can contain a simple numeric value such as 2500, a reference to another cell, or a complex expression such as (ab2*(@sqr(2))) or (12*ac24+52*11).

Other IF tests can be achieved by varying one of the preceding constructions. For example, these two statements are logically equivalent:

```
IF A =< B THEN C ELSE D
IF B > A THEN C ELSE D
```

Likewise, these two statements are equivalent:

```
IF A = B THEN C ELSE D
IF A <> B THEN D ELSE C
```

IF-THEN-ELSE Formulas

Every IF-THEN-ELSE statement can be broken into two separate parts—the IF test and its consequence. The first portion (for example, IF A=B) tests a logical condition. The second portion (for example, THEN C ELSE D) states the consequence of the test. The

THEN portion of the consequence is performed when the IF test is true, and the ELSE portion is performed when the IF test is false. Table 1 shows *SpeedCalc* formulas for the two IF tests described in the preceding section.

The consequence (THEN-ELSE) portion of the formula will always be the same expression—D+(C−D)*(...)—which represents the logical statement ELSE + (THEN − ELSE)*(...). When the ELSE portion of the consequence is to be 0, the expression reduces to a simple C*(...). When the THEN portion of the consequence is to be 0, all you need is the expression D − D*(...).

To express a complete IF-THEN-ELSE statement in a *SpeedCalc* formula, you need to multiply the consequence portion of the statement by the IF portion. For example, say that you wish to use this statement:

```
IF A>B THEN C ELSE D
```

The *SpeedCalc* equivalent is expressed by this formula:

```
D+(C−D) * @int((@sgn(A−B)+1)/2)
```

Note that we have placed the consequence portion—D+(C−D)—first and the IF portion—@int((@sgn(A−B)+1)/2)—last. The multiplication operator (*) separates the two portions of the statement.

Inside The IF Test

Recall that the computer ordinarily makes a decision based on an IF test by comparing two numbers. More specifically, it subtracts one number from the other, then determines whether the result is positive (true), or zero or negative (false). For example, to perform the statement IF A>B, we want to know whether the result of (A−B) is positive or not. If it is positive, then A is greater than B. If it is zero, then A equals B. If it is negative, A is less than B. In other words, after subtracting the two numbers, we then need to know the *sign* of the remainder.

SpeedCalc, of course, has no difficulty performing the subtraction. To determine the sign of the result, you need only enclose the expression in a @sgn( ) function, using the formula @sgn(A−B). When the result of A−B is positive, @sgn(A−B) resolves to 1. When the result of A−B is negative, it resolves to −1, and when the subtraction yields 0, @sgn(A−B) yields 0.

Now let's build on this basic expression to perform specific IF tests. To select only cases where A is greater than B (IF A>B), you need to select only the positive result. To do this, add the value of 1, divide by 2, and make the result an integer with the @int( ) function:

```
@int((@sgn(A−B)+1)/2)
```

This formula yields 1 when A is greater than B, and 0 in all other cases.

To select only cases where A is unequal to B (IF A<>B), you need to include negative as well as positive results (in other words, all nonzero results). The @abs( ) function easily converts any negative value into a positive value:

```
@abs(@sgn(A−B))
```

This formula yields 1 whenever A is unequal to B, and 0 only when A equals B.

Now we have formulas which resolve to the value 1 when the desired condition is true or the value 0 when it is false. Table 2 shows the complete formulas.

For both formulas in Table 2, when the IF test is true (resolves to 1), the cell is made equal to D+(C−D)*1. This performs the THEN part of the IF-THEN-ELSE statement, making the cell equal to C. When the IF test is false (resolves to 0), the cell is made equal to D+(C−D)*0. This performs the ELSE part of the IF-THEN-ELSE statement by making the cell equal to D.

To take a more realistic example, say that you want *SpeedCalc* to compute the equivalent of the following statement:

```
IF Q>9 THEN (this cell=) Q*P*.85 ELSE
   (this cell=) Q*P
```

Now assume that the value Q is in

Table 1: IF Formulas

IF Test	*SpeedCalc* formula
IF A > B	@int((@sgn(A−B)+1)/2)
IF A <> B	@abs(@sgn(A−B))

Table 3: Quantity Discounts

Unit price	$10.00
Quantity discounts:	1 to 9 - net
	10 to 99 -10%
	100 up -15%

Table 2: IF-THEN-ELSE Formulas

Logical expression	SpeedCalc Formula
IF A> B THEN C ELSE D	=D+(C−D)*@int((@sgn(A−B)+1)/2)
IF A<> B THEN C ELSE D	=D+(C−D)*@abs(@sgn(A−B))

Table 4: Quantity Discounts

Quantity	1	9	10	99	100	1000
Tot. list	10	90	100	990	1000	10000
Disc 10	0	0	10	99	0	0
Disc 100	0	0	0	0	150	1500
Tot amt	10	90	90	891	850	8500

Sample Spreadsheet

	AA	AB
001	price p	10.00
002	qty q	0.00
003	tot list	=ab1*ab2
004	disc 10	=ab3*.1*@int((@sgn(ab2−9)+1)/2) * @int((@sgn(100−ab2)+1)/2)
005	disc 100	=ab3*.15*@int((@sgn(ab2−99)+1)/2)
006	tot amt t	=ab3−ab4−ab5

cell AB1 and the value P is in cell AB2. This formula produces the desired result:

```
=ab1*ab2+(ab1*ab2*.85−ab1*ab2)*@int
  ((@sgn(ab1−9)+1)/2)
```

Boolean Operators

In certain cases the Boolean operator OR, NOT, or AND is required to perform a conditional test. The easiest of these to implement is NOT. If the value of A is 1, then the expression NOT A yields 0. If A equals 0, then NOT A equals 1. Both alternatives can be handled with this *SpeedCalc* expression:

```
abs(1−A)
```

The AND and OR operations can be simulated by combining two IF tests. For an AND operation, the results of both IF tests are multiplied:

```
[consequence] * [IF test 1] * [IF test 2]
```

For an OR operation, the results of both IF tests are added together:

```
[consequence] * ([IF test 1] + [IF test 2])
```

A Practical Illustration

For example, say that your business wants to calculate the quantity discounts diagrammed in Table 3. When you sell items in quantities of 9 or fewer, no discount is given. A 10 percent discount is given on purchases of 10 to 99 items, and purchases of 100 or more items qualify for a 15 percent discount.

To calculate the discounts within *SpeedCalc*, you need to set up a sheet with two conditional calculations; the first one requires an AND function. Run *SpeedCalc* and enter the sheet as shown in the figure.

To test whether the sheet performs as expected, enter some test results in cell AB2. You should get the results shown in Table 4.

Although the algorithms are simple, it is easy to make mistakes in logic when setting up such involved formulas. It often helps to write the statements on paper before entering the actual formula. Before using the formula for serious purposes, you should test it with some sample values to make sure it works correctly.

Amiga BASIC Style

Jim Butterfield, Associate Editor

Here's how to manage custom menus and output windows, read mouse input, trap background events, and master other techniques which give Amiga BASIC its unique character. The article also highlights some of the differences between Amiga BASIC and earlier BASICs, and includes a useful program for calculating mortgages.

There's a different style to BASIC programming on the Amiga. You should take a close look at new features; you'll discover concepts that lead to a radically different style of programming and user interaction.

To illustrate some of these concepts, let's construct a simple Amiga BASIC program which analyzes the five important variables in a home mortgage: principal (amount borrowed), interest rate, period of loan, monthly payment, and balance due. Since interest-compounding schedules are different in Canada than in the United States, the program includes an option for choosing either schedule. We'll discuss elements of the program as we go through it.

[Editor's note: In the following listing we have used the ◂ character to indicate the end of a program line. Don't try to type this character—we've deliberately chosen one that's not on the Amiga keyboard. The ◂ character merely shows where you should press RETURN to end one program line and start another.]

Initialization

```
REM Mortgage (Version 1)◂
DIM title$(6),site$(2),pudef$(5)
,value#(5),peryear(2),compound(2
)◂
cal=4:site=1◂
```

The REM identifies the program and version. The DIM statement defines the six arrays used in the program, which we'll discuss as we go along. Note that there are no line numbers in Amiga BASIC. They are not needed. Even with GOTO or GOSUB, it's usual to identify a line with a label, not a number. (You may include line numbers if you like—a feature included for the sake of compatibility with other BASICs—but since the line numbers are treated simply as labels, numeric order is irrelevant.)

Also, notice that we use descriptive words for variable names.

In the versions of BASIC on earlier Commodore computers, only the first two characters of the variable name were significant (HO$ and HOUSEHOLD$ would be considered the same name). In Amiga BASIC, names can be up to 40 characters long with every character significant (Householdbudget1 and Householdbudget2 are recognized as distinct names). Descriptive variable names make the program much easier to understand and reduce the need for explanatory REM statements. We also set the default value of the two variables that determine which menu items are selected. The loan variable to be calculated (*cal*) is 4, the payment amount. The default interest compounding schedule (*site*) is that for country 1, Canada. Change either of these if you wish.

```
DATA Principal,Rate,Years,Paymen
t,Balance,Quit←
MENU 5,Ø,1,"Calculate"←
FOR j=1 TO 6:READ title$(j)←
MENU 5,j,1-(j=cal),"  "+title$(j
):NEXT j←
```

The DATA statement contains the items for the first of our custom menus, as well as the captions for the output window (the array *title$*). One of the most significant features of Amiga BASIC is that the programmer can easily construct custom menus.

We'll choose menu 5 for our first custom menu so that menus 1–4 can retain their default uses: Project, Edit, Run, and Windows. The first MENU statement sets Calculate as the title for the menu, then the FOR-NEXT loop reads the DATA items into the corresponding menu slots. Note the expression *1−(j=cal)* for the third parameter of the MENU statement in the loop. Just like earlier Commodore BASICs, Amiga BASIC interprets a true expression as −1 and a false expression as 0, so 1−(j=cal) will evaluate to 1−(−1) = 2 when the value of *j* equals the value of *cal*, and 1−(0) = 1, otherwise. A value of 2 for this parameter puts a check to the left of the menu item, so this feature is used to indicate which calculation option is currently selected. A value of 1 displays the menu item without a checkmark, but still makes it active; a value of 0 would deactivate the menu item, leaving it dimmed, or *ghosted*, and impossible to select.

```
DATA Canada,2,6,USA,12,1←
MENU 6,Ø,1,"Country"←
FOR j=1 TO 2:READ site$(j),perye
ar(j),compound(j)←
MENU 6,j,1-(j=site),"  "+site$(j
):NEXT j←
```

Different rules are used in the U.S. and Canada to work out a monthly interest rate based on the annual interest figure. In the U.S., the annual amount is simply divided by 12. In Canada, semiannual compounding is used, which involves dividing by two to get the semiannual rate and then using a more complex formula. The user will be able to pick the appropriate system from menu 6, which is titled Country. It would not be too hard to add extra menu items, such as compounding quarterly (the numeric DATA items would be 4,3). The FOR-NEXT loop here uses the same technique for flagging the current menu selection as the one above.

Format With PRINT USING

```
DATA "#,###,###.##"←
DATA "    ###.###% "←
DATA "    ###.###  "←
DATA "#,###,###.##"←
DATA "#,###,###.##"←
FOR j=1 TO 5:READ pudef$(j):NEXT
j←
```

These are the PRINT USING templates that tell how the numeric values of the five loan variables are to be printed. The principal amount, for example, is printed as a dollars-and-cents value. The annual interest rate, in contrast, will be shown to three decimal places with a percent sign.

```
DATA 1ØØØØ,1Ø,1Ø,Ø,Ø←
FOR j=1 TO 5:READ value#(j):NEXT
j←
```

These are just arbitrary figures to appear on the initial screen. I've picked a principal amount of $10,000 at 10 percent over ten years. You could substitute your own default values if you like. Once the program is running, any of these values can easily be changed.

An important point: Note that the array into which the values are read, *value#*, has an extra symbol at the end. The # sign (pound sign, hash mark, or whatever you want to call it) indicates that these variables are *double precision*. If you've worked with previous Commodore machines which offered only one level of numeric precision, you might be unclear about this issue. Here's the story: In earlier Commodore BASICs, variables worked to about ten digits of accuracy. That was enough—just barely enough—to do most home finance calculations. Normal (single-precision) Amiga BASIC variables—the type you usually get if you don't add a type identifier after the variable name—are reliable to only about seven digits. This means that it can't handle amounts of over about $167,000 without losing pennies.

Computer scientists will tell you that single-precision Amiga BASIC variables have a 24-bit *mantissa*, as opposed to the 32-bit mantissa in earlier Commodore BASICs. What it means to you is this: Whenever you need to deal with dollars-and-cents values—or with other values requiring a high accuracy—you need to call for a double-precision variable. Such a variable will have more accuracy—enough to cover a federal budget and still be exact on the pennies. To specify double precision, add a # sign to the end of the variable name. Be careful to include the sign each time you use the variable name, however. Amiga BASIC will consider *value* and *value#* to be two different variables.

A Custom Window

```
WINDOW 2,"Mortgage",(1Ø,1Ø)-(4ØØ
,1ØØ),8←
WINDOW OUTPUT 2←
GOSUB calc:GOSUB showval←
LOCATE 7,1←
PRINT "Use menu buttons to selec
t option."←
PRINT "Click on existing values
to change."←
GOSUB hang←
WINDOW CLOSE 2←
END←
```

Now we open a new window in which the calculations will appear. The only gadget we put on the window is the closing gadget (code 8). It's there so that the user can still put away the window manually in case the program is stopped. The window is not only created, but also selected for output. Then the initial calculations are displayed, along with brief instructions near the bottom of the window.

The program's main job is a subroutine called *hang*. We'll stay in that subroutine until the user wants to quit, at which time the window will be closed. Here is the *hang* subroutine:

```
hang:◂
ON MENU GOSUB event◂
ON MOUSE GOSUB event◂
MOUSE ON◂
MENU ON◂
kwit=Ø◂
WHILE kwit<>1:WEND◂
MOUSE OFF◂
MENU OFF◂
MENU RESET◂
RETURN◂
```

We define an action for the mouse and for the menus we previously defined. Clicking the left mouse button or selecting a menu item invokes the *event* subroutine. These two activities are *interrupts* or *event traps*. After they are activated with MENU ON and MOUSE ON, they will remain in place, waiting for the appropriate event to happen, until they are canceled or turned off. While they are active, it doesn't matter what the program is doing; a suitable stimulus will immediately cause the program to jump to the specified subroutine.

A variable called *kwit* is used by the program to tell when it's time to quit. As long as it's zero, the program stays in the WHILE–WEND loop. How does it ever get out of this seemingly endless loop? Remember the event traps we just enabled. Pressing the left mouse button or selecting a menu item will trigger a GOSUB to the *event* routine, which in turn calls subroutines to process the button click or menu selection. One menu selection, the *Quit* option from the Calculate menu, will change the value of *kwit* to one to end the loop. After exiting the loop, we'll shut off the menu and mouse, disconnect the event traps, and return to the main program which ties things up.

A Major Event

```
event:◂
ms=MOUSE(Ø):mn=MENU(Ø)◂
IF mn THEN GOSUB menuhit◂
IF ms THEN GOSUB eek◂
IF kwit=Ø THEN GOSUB calc:GOSUB
showval◂
RETURN◂
```

Now let's look at the routine where the real action takes place. When we arrive at the *event* subroutine, we know that one of two things has happened. Either the left mouse button has been clicked or a menu item has been selected by using the right mouse button. The MOUSE and MENU functions are used to check which, and the appropriate service subroutine is called. Once the new value for *cal* or *site* has been established, we're ready to calculate new values, but first we check that *kwit* is still zero—we don't want to calculate values if the Quit option from the Calculate menu was selected. The new financial values are determined by calling the subroutine *calc*, then displayed using the *showval* subroutine. Keep in mind that we'll come back to this routine to recalculate anytime the data elements—or the rules—are changed.

```
calc:◂
ON ERROR GOTO oops◂
principal#=value#(1)◂
r1#=(value#(2)/1ØØ/peryear(site)
+1)^(1/compound(site))◂
rate#=r1#-1◂
months=value#(3)*12◂
payment#=value#(4)◂
balance#=value#(5)◂
ON cal GOSUB fprin,fintr,fper,fp
ay,fbal◂
scale=1ØØ:IF cal=2 OR cal=3 THEN
scale=1ØØØ◂
value#(cal)=INT(value#(cal)*scal
e+.99)/scale◂
ON ERROR GOTO Ø◂
RETURN◂
```

The *calc* subroutine is where the dirty work begins. The principal, interest rate, number of periods, payment amount, and final balance are extracted from the *value#* array so that they can be used by the various calculation programs more easily. Note that in most cases, we retain double-precision accuracy with the # sign. The monthly interest rate is worked out by a fairly complex formula, and the number of months equals the number of years times 12.

The variable *cal* tells us what to calculate. Depending on its value, we'll call *fprin* (find principal), *fintr* (find interest rate), *fper* (find period), *fpay* (find payment), or *fbal* (find balance). The calculation with *scale* rounds any calculated value to the next highest penny, or, if not a money figure, to three decimal places.

The calculation subroutine also includes an error trap, since some calculations are impossible or ridiculous (for example, how long would it take to pay off a $1,000 mortgage with a payment of $0 per month?). Problems are directed to an event trap named *oops*.

```
oops:◂
value#(cal)=Ø◂
RESUME oops2◂
oops2:◂
WINDOW 2◂
RETURN◂
```

If there's any calculation problem, we set the calculated value to zero and give up. We do not go back to the detailed calculation program. Instead, using *oops2*, we return to the main *calc* routine. But, first, it's necessary to reopen WINDOW 2, since the Amiga always closes any secondary windows when an error occurs. Notice that the message at the bottom of the window is not reprinted. So if you see the window blink, then reappear minus the message and with the value being calculated set to zero, an error has been trapped. If this occurs when you enter what seem to be legitimate values, it may indicate that you made an error while entering the program. For this reason you may want to omit the ON ERROR statements until you are confident that you have eliminated all typing mistakes in the program.

Here are the five calculation routines. We won't plunge into details of the math here, since it's rather complex.

```
fprin:◂
value#(1)=(balance#+payment#*(r1
#^months-1)/rate#)/r1#^months◂
RETURN◂
◂
fintr:◂
rØ#=Ø:r1#=EXP(75/months):IF r1#>
2 THEN r1#=2 ◂
rate#=r1#-1:r9#=rate#*1ØØ◂
pØ#=balance#+payment#*months-pri
ncipal#◂
p9#=(balance#+payment#*(r1#^mont
hs-1)/rate#)/r1#^months-principa
l#◂
IF pØ#<Ø OR p9#>Ø THEN ◂
r2#=Ø◂
ELSE◂
flop%=Ø◂
WHILE ABS(r9#-rØ#)>.ØØ1◂
flop%=1-flop%◂
IF flop%>Ø THEN◂
r2#=(rØ#+r9#)/2◂
ELSE◂
r2#=rØ#-pØ#*(r9#-rØ#)/(p9#-pØ#)◂
END IF◂
r1#=(1+r2#/1ØØ/peryear(site))^(1
/compound(site))◂
rate#=r1#-1◂
p2#=(balance#+payment#*(r1#^mont
hs-1)/rate#)/r1#^months-principa
l#◂
IF p2#>Ø THEN◂
rØ#=r2#:pØ#=p2#◂
ELSE◂
r9#=r2#:p9#=p2#◂
END IF◂
WEND◂
END IF◂
value#(2)=r2#◂
```

```
RETURN◂
◂
fper:◂
value#(3)=LOG((payment#-rate#*ba
lance#)/(payment#-rate#*principa
l#))/LOG(rl#)/12#◂
RETURN◂
◂
fpay:◂
value#(4)=rate#*(principal#*rl#^
months-balance#)/(rl#^months-1)◂
RETURN◂
◂
fbal:◂
value#(5)=principal#*rl#^months-
payment#*(rl#^months-1)/rate#◂
RETURN◂
```

The only one of the above routines that's lengthy is *fintr*. There's no simple formula for the interest rate, so we must zero in on the correct value by repeated calculations.

Displaying Results

Now to display the calculated values:

```
showval:◂
FOR j=1 TO 5◂
LOCATE j,1◂
IF j=cal THEN ◂
PRINT "*";◂
ELSE◂
PRINT " ";◂
END IF◂
PRINT title$(j);SPACE$(2Ø)◂
LOCATE j,12◂
PRINT USING pudef$(j);value#(j)◂
NEXT j◂
RETURN◂
```

For a good human interface, I wanted to distinguish between the calculated item and the entered values. The title for the value being calculated will be preceded by an asterisk. SPACE$ is used to generate a string of blanks to wipe out any old values.

A Choice Is Made

```
menuhit:◂
ms=Ø◂
IF mn>4 THEN◂
mnl=MENU(1)◂
ON mn-4 GOSUB newcalc,style◂
END IF◂
RETURN◂
```

Here's the routine to handle menu selections. The value *mn*, given the value of MENU(0) in the calling routine, is used to determine which menu is involved. MENU(1) tells us which item from the menu has been selected. We then subtract 4 from *mn* to get an offset of 1 or 2 for the ON-GOSUB statement.

```
eek:◂
x=MOUSE(3):y=MOUSE(4)◂
IF x>5 AND x<19Ø THEN◂
v=INT((y+8)/8)◂
IF v>Ø AND v<6 AND v<>cal THEN◂
LOCATE v,12:PRINT SPACE$(2Ø)◂
LOCATE v,12:INPUT value#(v)◂
LOCATE v,12:PRINT USING pudef$(v
);value#(v)◂
END IF◂
END IF◂
RETURN◂
```

The *newcalc* subroutine is called when menu 5, the Calculate menu, is selected. If the item selected from that menu is 1–5, the previously selected menu item has its checkmark removed, and a checkmark is placed beside the newly selected item. The value of *cal* is updated to show which variable is now being calculated. If menu item 6, Quit, was chosen, we instead set the value of *kwit* accordingly. The *style* subroutine sets *site* to the selected country when an item is selected from menu 6, the Country menu.

```
newcalc:◂
IF mnl<6 THEN◂
MENU 5,cal,1◂
cal=mnl◂
MENU 5,cal,2◂
ELSE◂
IF mnl=6 THEN kwit=1◂
END IF◂
RETURN◂
style:◂
IF mnl<3 THEN◂
MENU 6,site,1◂
site=mnl◂
MENU 6,site,2◂
END IF◂
RETURN◂
```

When the left mouse button is clicked, the *eek* subroutine allows entry of a new value. It's important to read MOUSE(0) before reading the mouse's position, but in this case, that's already been done in the *event* routine that calls *eek*. The *x* and *y* coordinates of the mouse pointer's current position come from MOUSE(3) and MOUSE(4), since those functions return the position of the mouse when the button was clicked. MOUSE(1) and MOUSE(2) return the mouse's position at the time of the MOUSE(0) call, so either would probably give comparable results in this case. Remember that we are reading pixel positions, not character positions. Before recognizing a click as a request to enter input, we check that the pointer was reasonably close to one of the displayed values. One more limitation is that we won't allow an entry for the *cal* variable: The computer calculates that value.

Once we know it's a valid variable, we clear the old value using SPACE$, input a new value, and then print it neatly formatted in the space provided.

Maiden Voyage

Let's give the program a trial run. You'll see the window appear. If you have used the initial values suggested, you'll notice that the program has calculated a payment of $131.04. That's the Canadian computation. Now press the right button, slide the mouse pointer up to the Country menu, and move down to *USA* before you release the button. The payment should change to $132.16.

This is a ten-year mortgage. Let's see what the balance would be after five years. Use the right button (also called the menu button, for obvious reasons) to select the Balance option from the Calculate menu. The balance will show a slightly negative amount. That's okay (each payment is rounded up a fraction of a penny, so the final payment will be slightly less than zero). Next, move the pointer up to the Years value in the display window menu and click the left button. The computer is inviting you to enter a new value: Enter 5 for five years. Observe that the balance still due after five years is a little over $6,000.00.

How long to pay it off at $150 a month? Select Years from the Calculate menu. Change the Balance value to 0 and the Payment value to 150. The answer is a little over eight years. If you change the interest rate to 12 percent, you'll see that it would take over nine years to pay off the loan. At 18 percent, you wouldn't live long enough to pay it off at $150 a month, and at 20 percent, it's impossible (note the Years value is set to zero to indicate the error). When you've snooped through the combinations enough to satisfy yourself, select Quit. And don't forget to save the program. If your answers don't match these, check the formulae for typographical errors.

After running through this exercise, think how different things would be on any eight-bit computer. It's not just the mortgage calculation; it's the style of the machine. With a fresh approach, you can make your Amiga more flexible and useful than any computer you've used before. ©

Home Financial Calculator For Atari ST

Patrick Parrish, Programming Supervisor

Rarely has there been a program integrating as many useful loan and investment features as "Home Financial Calculator." It is versatile, easy to use, and flexible. Rapid recalculation features make it an ideal tool for "what if" projections. A calculator mode with memory lets you solve problems not directly supported by the program, and you can pass values generated by one calculation to another. Home Financial calculator was originally published in the May 1985 issue of COMPUTE!. *This new version is for any Atari ST computer which has TOS in ROM.*

"Home Financial Calculator" integrates a number of common financial calculations in a menu-driven package. It also features a calculator mode or scratch pad area where program variables can be manipulated using common mathematical operations.

Be particularly careful when typing the long lines in this program which contain financial formulae. A mistyped program may still run, but the results it gives could be inaccurate.

When you run the program, a main menu offers you a choice of Investment or Loan calculations. Type I or L to reach the appropriate submenu.

Easy "What If" Projections

Before looking at any calculations, let's consider some basics of the program. Home Financial Calculator uses some parameters or variables repeatedly in the calculations. These variables are *Total* (also referred to as Future Value, Total Owed, and so forth, depending on the calculation); *Present Value* (principal); *Interest Rate*; *Years*; *Months*; *Number of Periods* (of either compounding, deposits, withdrawals, or payments, depending on the application); *Deposits*; and *Withdrawals*. When in the calculator mode (explained below), you'll reference these eight variables with the single letters *T, P, I, Y, M, N, D,* and *W*.

As you work with Home Financial Calculator, the values of the eight variables are preserved until you change them. Whenever the program asks you for an input (for example, Interest), the current value of that variable is displayed (zero if no value has been entered yet). If you want to keep the current value, just press Return. Otherwise, enter the new value and press Return.

With this feature, Home Financial Calculator makes it easy for you to generate "what if" projections. Simply run the same calculation repeatedly, each time changing a previously entered value. Press Return to keep a value, and change only one or two values to see the effect on the final result.

You can also store the current value into the calculator mode's Memory Register or recall a value from the Memory Register. To see how all this works, let's take a closer look at your options.

Your Investment Menu

Here is the Investment submenu that appears when you type I from the main menu:

1) Future Value with Periodic Interest
2) Future Value with Interest Compounded Continuously
3) Future Value with Regular Deposits
4) Future Value with Cash Flows
5) Withdrawal of Funds
6) Net Present Value
7) Calculator Mode
8) Return to Main Menu.

Determine which option you want and press the appropriate key.

Each option displays screen prompts which ask you to input several values. These values are stored in the eight variables mentioned above: *T* for Total (Future Value), *P* for Present Value (principal), *I* for Interest Rate, *Y* for Years, *M* for Months, *N* for Number of Periods, *D* for Deposits, and *W* for Withdrawals. Of course, not all calculations require you to enter all these values, while others may ask for additional information.

Most calculations can be solved for any *one* of the variables. To solve for a variable, enter an uppercase X at the corresponding input prompt. For example, you could enter values for everything except the Interest Rate, typing X at the Interest Rate prompt. Home Financial Calculator then solves for the Interest Rate.

Remember, however, that the program can solve for only *one* variable during each calculation. If you enter an X at more than one prompt, the program does not have enough information to calculate an answer.

Future Value With Periodic Interest

Home Financial Calculator's options are fairly self-explanatory when you run the program, but let's try an example. We'll calculate the future value of an investment drawing periodic interest. This kind of investment could be a savings account, interest-bearing checking account, bonds, or a money market account. Choose this option by entering 1 at the Investment submenu.

After the screen clears, the program asks for the first input—Future Value, which appears with an asterisk (*). Below this is a zero (the current value of this variable in memory; all variables start out with a value of zero). Following this is an input prompt.

The asterisk preceding Future Value means that this is one of the variables you can solve for. (A variable *not* preceded by an asterisk means that variable *cannot* be solved for in that particular calculation, so X would be an illegal response.) If you'd like to calculate the Future Value, enter an X here, and answer all the other prompts with the appropriate values.

Let's calculate the future value of a $1,000 investment drawing 8 percent interest for two years and three months, with four compounding periods each year. Enter an X for *Future Value*, since we'll be solving for this total. Answer *Present Value* with 1000 (the principal you're investing); *Annual Int Rate (%)* with 8 (enter the percentage, not a fraction); *For # Of Years* with 2; *For # Of Months* with 3; and *# Of Periods (Compounding)* with 4. After you enter the last value, Home Financial Calculator figures the *Total Future Value* and displays the answer—$1195.09.

Now suppose you wish to know the future value of the same $1,000 investment if you make 9 percent interest. Choose option 1 on the Investment submenu again and rerun the calculation. Notice how Home Financial Calculator automatically prints the current value of each variable at each prompt. The *Future Value* prompt shows a current value of 1195.09 from the previous calculation. Type an X at this prompt, 9 for Interest Rate, and Return at all other prompts to preserve their values. The result should be $1221.71.

The versatility of Home Financial Calculator becomes apparent when you realize how many different ways you can run this calculation. Using this same menu option, you can calculate the initial investment (or present value) necessary to accrue a certain future value with periodic interest; the interest rate necessary to accrue a future value from a present value; or the time (in years and months) it would take to accumulate a future amount from an initial investment with periodic interest payments. Just enter an X for the unknown value you're seeking and fill in all the other prompts.

Future Value With Interest Compounded Continuously

Option 2, a variation of option 1, handles investments paying a continuous interest rate. Like option 1, option 2 can handle a number of calculations—just place an X in the slot you'd like to solve for.

Here, after entering all other parameters, you can calculate the future value of an investment; the initial investment required to reach a certain future value; the interest required to reach a desired future value; or the time required to reach a certain future value at a specified interest rate.

Notice that any variables used in option 1 will be displayed with their current values when running option 2. Recall that the eight major variables in Home Financial Calculator retain their values throughout the program until you change them. This feature is convenient when going from one option to another on the Investment or Loan submenus.

In addition, the values are preserved for use in the calculator mode. For instance, you could compare the effect of continuously compounded interest to periodic interest (option 1) without having to retype the input.

Future Value With Regular Deposits

If you're interested in setting up an annuity, you'd choose option 3 on the Investment submenu. You can determine the future value of an account (such as a savings account, Individual Retirement Account, or college or vacation fund) with regular deposits where interest is compounded with each deposit.

Option 3 can also tell you the amount of each deposit necessary to accrue a future value; the interest rate needed to provide some future value with regular deposits; or the time it would take to amass a future value with regular deposits.

Future Value With Cash Flows

Option 4 does a single calculation—it always solves for *Future Value*, so don't enter an X anywhere. It calculates the future value of an investment with yearly cash flows (either positive or negative). The *Annual Interest Rate* you input here is the growth rate on the money you've invested.

As an example, suppose you wish to determine the value of a vacation fund collected over four years. You're asked for the number of years, then for the deposit or withdrawal each year. You deposit $500 in the fund the first year and $200 the second. The third year you are forced to withdraw $300 (entered as −300), and the fourth year, you put in $400. The fund has a growth rate of 12 percent. Its value after four years will be $1,017.34.

A future value determination can also tell you whether an investment is worthwhile. If the future value of all cash flows is positive or zero, the investment is profitable. A negative future value, on the other hand, represents a losing investment.

Withdrawal Of Funds

If you intend to open an account from which you can regularly withdraw funds, choose option 5. With this option, you can determine the initial deposit required in the account to cover your withdrawals; the amount you can withdraw regularly from this account; the rate of interest you must make on funds in the account; or the period of time over which you can make withdrawals.

Net Present Value

Option 6 lets you determine the feasibility of a prospective investment by calculating its net present value. Net present value is the current value of all future yearly cash flows to an investment along with any initial cash requirement. The interest rate you input here is the rate of return you require on your investment. A positive net present value indicates a profitable investment, while a negative result signifies a losing investment.

As an example, suppose you have the opportunity to make a $2,000 investment which would return $1,500 the first year, cost you $750 the second year, and return $1,900 the third year. You hope to make 13 percent on your money. With option 6, you would determine a net present value of $56.87, representing a profitable investment.

The Calculator Mode

Option 7 puts you in the calculator mode (also available from the Loan submenu). Calculator mode works very much like a hand-held calculator with a single memory. You can type in a value or recall one from a variable by entering its symbol—T(otal), P(resent Value), I(nterest Rate), Y(ears), M(onths), N(umber of Periods), D(eposits), and W(ithdrawals). You can perform simple math on values stored in the Memory Register using reverse Polish notation. And you can use the results in future calculations.

When you enter calculator mode, the calculator command line appears on the screen:

```
V S H R M+ M− M* M/ MR MC
MEM=0
```

Here are the commands:

V	View the values of the eight primary variables
S	Store Memory Register into a variable
H	Help—prints the command line
R	Return to main menu, exit calculator mode
M+	Add the last input to the Memory Register
M−	Subtract the last input from the value in the Memory Register and store the result in the Register
M*	Multiply the last input times the value in the Memory Register and store the result in the Register
M/	Divide the last input into the value in the Memory Register and store the result in the Register
MR	Memory Recall
MC	Memory Clear to zero
MEM=	Memory Register's current value

If you've run through a sample investment calculation, you now have some variables in memory. Enter V in the calculator mode to see them. The screen displays the eight values currently in memory for the eight variables.

To work with one of these variables, enter one of their letters (*T, P, I, Y, M, N, D*, or *W*) and press Return. Then type M+ to add it to the Memory Register (all variables must be stored in the Register before you can perform any operations on them). Suppose you put the current value for T into the Register and now wish to add $229 to this value. Enter 229, press Return, then type M+ and press Return. The addition is performed and the result displayed. To store this value back into the T variable, enter S for Store. A prompt appears, requesting the variable in which you intend to store the value. Type T to store the value into the variable T.

You can also use the Memory Register to hold a value not represented by any of the eight variables. To do this, determine a value using the calculator mode and store it into the Memory Register with M+. Then, when you're running a calculation elsewhere in the program, you can substitute this value for any of the eight primary variables by typing MR (Memory Recall) at the appropriate prompt. MR can be used both in the calculator mode and at any prompt where the previous value is displayed.

Finally, option 8 on the Investment submenu returns you to the main menu. Once there, you can perform some loan calculations by typing L.

Loan Calculations

Here is the Loan calculations submenu:

1) Regular Loan Payments
2) Remaining Loan Liability
3) Final Loan Payment
4) Single Payment Loan
5) Loan Amortization Schedule
6) Calculator Mode
7) Return to Main Menu

Regular Loan Payments

Option 1 handles a number of calculations for equal payment loans. You can figure the principal of a loan; the amount of each regular payment necessary to repay a loan; the annual interest rate on a loan with regular payments; or the term of the loan.

Remaining Loan Liability

With option 2, you can determine the remaining balance on a loan with regular payments after a number of payments have been made. Enter the principal on the loan, the amount of each payment, the annual interest rate, the number of payments yearly, and the last payment number.

Final Loan Payment

Option 3 calculates the amount of the final payment on a loan. In many cases, the last payment of a loan will vary from the amount of the regular payment. This option handles situations where the final payment is greater than ("balloon payments") or less than the regular payment.

Single Payment Loan

Option 4 calculates the amount owed on a loan that is paid off with a single payment. You must input the principal on the loan, its annual interest rate, its term in years and months, and the number of times a year the interest on the principal is compounded.

Loan Amortization Schedule

Option 5 displays a loan amortization schedule. Enter the principal on the loan, the amount of each payment, the annual interest rate, the term of the loan, and the number of payments yearly. Then enter the period of the year in which the loan began (for instance, 10 for October) and the range in years of the amortization schedule you'd like to examine.

Because of the complexity of these calculations, there may be a delay before the output appears on the screen, especially if you have chosen to look at the latter years in a long-term loan repayment schedule (such as a home mortgage). When the amortization table appears, it displays the payment number, the beginning balance for the period, the amount paid toward the loan principal, the amount paid in interest, and the ending balance. To keep the information from scrolling off the screen, the program shows only a few payment periods at a time. Press Return to view another screenful. When the end of a year is reached, the program gives the total amounts paid on the principal and in interest for the year. In addition, when the last period of the loan is reached, the program displays the final payment for the loan.

The last two options on the Loan submenu are the same as those on the Investment submenu.

Modifying The Program

Home Financial Calculator is written in a modular format for easy modification. For many routines, it uses common input labels (lines 4590–4960) and some output labels (lines 4970–5050). If you want to add an investment or loan calculation routine, choose the labels from these lines that fit your application.

Also, you may wish to add a printer option to the loan amortization schedule. Examine lines 3140–3840. Here, variable D5 (defined in line 140) determines the number of loan payments considered on each screen. Variables S1, S2, S3, and S4 (defined in lines 150–180) format the output horizontally on the screen.

Home Financial Calculator For Atari ST

Version by George Miller, Assistant Technical Editor

```
1Ø      GOSUB 534Ø
2Ø      RES = PEEK(SYSTAB+Ø)
3Ø      IF RES <> 4 THEN 6Ø
4Ø      ?"Please switch to Medium
         or High"
5Ø      ? "Resolution.":STOP
6Ø      COLOR 1,1
7Ø      OIM V(8)
BØ      V$="TPIYMNOW"
9Ø      C$="VSHR"
1ØØ     CØ$="V S H R "
11Ø     C1$="M+ M- M* M/ MR MC"
12Ø     C2$="M+M-M*M/MRMC"
13Ø     Q$=""
14Ø     O5=12
15Ø     S1=1Ø
16Ø     S2=25
17Ø     S3=4Ø
18Ø     S4=55
19Ø     TITLE$=" Home Financial C
        alculator "+CHR$(Ø)
2ØØ     OOSU8 534Ø:OOSU8 TITLE8AR

21Ø     PRINT "INVESTMENTS OR LOA
        NS? (Select 'I' or 'L')"
22Ø     A$=CHR$(INP(2))
23Ø     IF A$="I" OR A$ = "i" THE
        N 26Ø
24Ø     IF A$="L" OR A$ = "1" THE
        N 212Ø
25Ø     OOTO 22Ø
26Ø     OOSU8 534Ø
27Ø     TITLE$=" INVESTMENTS ":OO
        8U8 TITLE8AR
28Ø     OOTOXY 1Ø,5:PRINT "1) FUT
        URE VALUE WITH PERIOOIC I
        NTEREST"
29Ø     OOTOXY 1Ø,6:PRINT "2) FUT
        URE VALUE WITH INTEREST C
        OMPOUNOEO CONTINUOUSLY"
3ØØ     GOTOXY 1Ø,7:PRINT "3) FUT
        URE VALUE WITH REGULAR OE
        POSITS"
31Ø     GOTOXY 1Ø,8:PRINT "4) FUT
        URE VALUE WITH CASH FLOW8
        "
32Ø     GOTOXY 1Ø,9:PRINT "5) WIT
        HORAWAL OF FUNOS"
33Ø     GOTOXY 1Ø,1Ø:PRINT "6) NE
        T PRESENT VALUE"
34Ø     GOTOXY 1Ø,11:PRINT "7) CA
        LCULATOR MOOE"
35Ø     GOTOXY 1Ø,12:PRINT "8) RE
        TURN TO MAIN MENU"
36Ø     GOTOXY 1Ø,14:PRINT "YOUR
        CHOICE?";
37Ø     A=INP(2)-4B
38Ø     IF A<1 OR A>8 THEN 37Ø
39Ø     ON A GOTO 42Ø,6BØ,92Ø,131
        Ø,15ØØ,1B9Ø,4ØØ,19Ø
4ØØ     GOBUB 4Ø6Ø
41Ø     GOTO 19Ø
42Ø     GOSU8 534Ø
43Ø     TITLE$=" FUTURE VALUE WIT
        H PERIOOIC INTEREBT ":GO8
        UB TITLE8AR
44Ø     PRINT
45Ø     GOSU8 459Ø
46Ø     GOSU8 463Ø
47Ø     PRINT "*";
48Ø     GOSUB 472Ø
49Ø     PRINT "*";
5ØØ     GOSU8 476Ø
51Ø     IF E=4 THEN 53Ø
52Ø     GOSU8 48ØØ
53Ø     GOSU8 485Ø
54Ø     IF E<>1 THEN 57Ø
55Ø     V(1)=INT(V(2)*(1+V(3)/V(6
        ))^(V(6)*Y)*1ØØ+.5)/1ØØ
56Ø     GOSU8 497Ø
57Ø     IF E<>2 THEN 6ØØ
58Ø     V(2)=INT(V(1)/((1+V(3)/V(
        6))^(V(6)*Y))*1ØØ+.5)/1ØØ

59Ø     GOSU8 5ØØØ
6ØØ     IF E<>3 THEN 63Ø
61Ø     V(3)=INT((V(6)*(V(1)/V(2)
        )^(1/(V(6)*Y))-V(6))*1ØØØ
        Ø+.5)/1ØØØØ
62Ø     OOSUB 5Ø3Ø
63Ø     IF E<>4 THEN 66Ø
64Ø     V(4)=LOO(V(1)/V(2))/(V(6)
        *LOG(1+V(3)/V(6)))
65Ø     OOSU8 5Ø6Ø
66Ø     GOSU8 521Ø
67Ø     GOTO 26Ø
6BØ     GOSUB 534Ø
69Ø     TITLE$=" FUTURE VALUE WIT
        H INTEREST COMPOUNOEO CON
        TINUOUSLY ":OOSUB TITLEBA
        R
7ØØ     PRINT
71Ø     OOSU8 459Ø
72Ø     OOSU8 463Ø
73Ø     PRINT "*";
74Ø     GOSU8 472Ø
75Ø     PRINT "*";
76Ø     OOSUB 476Ø
77Ø     IF E=4 THEN 79Ø
78Ø     OOSU8 4BØØ
79Ø     IF E<>1 THEN 82Ø
8ØØ     V(1)=INT(V(2)*EXP(V(3)*Y)
        *1ØØ+.5)/1ØØ
81Ø     GOSU8 497Ø
82Ø     IF E<>2 THEN 85Ø
83Ø     V(2)=INT(V(1)/EXP(V(3)*Y)
        *1ØØ+.5)/1ØØ
84Ø     GOSUB 5ØØØ
85Ø     IF E<>3 THEN 88Ø
86Ø     V(3)=INT(LOO(V(1)/V(2))/Y
        *1ØØØØ+.5)/1ØØØØ
87Ø     GOSU8 5Ø3Ø
BBØ     IF E<>4 THEN 66Ø
89Ø     V(4)=INT(LOG(V(1)/V(2))/V
        (3)*1ØØ+.5)/1ØØ
9ØØ     GOSU8 5Ø6Ø
91Ø     GOTO 66Ø
92Ø     GOSU8 534Ø
93Ø     TITLE$=" FUTURE VALUE WIT
        H REGULAR OEPOSIT8 ":GOSU
        B TITLE8AR
94Ø     PRINT
95Ø     GOSUB 459Ø
96Ø     PRINT "*REGULAR OEPOSIT $
        "
97Ø     C=6
98Ø     GOSUB 385Ø
99Ø     PRINT "*";
1ØØØ    GOSU8 472Ø
1Ø1Ø    PRINT "*";
1Ø2Ø    GOSUB 476Ø
1Ø3Ø    IF E=4 THEN 1Ø5Ø
1Ø4Ø    GOSUB 48ØØ
1Ø5Ø    GOSU8 485Ø
1Ø6Ø    IF E<>1 THEN 1Ø9Ø
1Ø7Ø    V(1)=INT(V(7)*V(6)*((1+V(
        3)/V(6))^(V(6)*Y)-1)/V(3)
        *1ØØ+.5)/1ØØ
1ØBØ    GOSU8 497Ø
1Ø9Ø    IF E<>3 THEN 123Ø
11ØØ    V(3)=.99
111Ø    I=Ø
112Ø    T=INT(V(7)*(((1+V(3)/V(6)
        )^(V(6)*Y)-1)/(V(3)/V(6))
        )*1ØØ+.5)/1ØØ
113Ø    TE=ABS(V(3)-I)/2
114Ø    I=V(3)
115Ø    IF ABB(T-V(1))/V(1)<.ØØØØ
        5 THEN 121Ø
116Ø    IF T<V(1) THEN 119Ø
117Ø    V(3)=V(3)-TE
118Ø    GOTO 112Ø
119Ø    V(3)=V(3)+TE
12ØØ    GOTO 112Ø
121Ø    V(3)=INT(V(3)*1ØØØØ+.5)/1
        ØØØØ
122Ø    GOSUB 5Ø3Ø
123Ø    IF E<>4 THEN 126Ø
124Ø    V(4)=LOG(V(3)*V(1)/(V(6)*
        V(7))+1)/(V(6)*LOG(1+V(3)
        /V(6)))
125Ø    GOSUB 5Ø6Ø
126Ø    IF E<>7 THEN 66Ø
127Ø    V(7)=INT(V(1)*(V(3)/V(6))
        /((1+V(3)/V(6))^(V(6)*Y)-
        1)*1ØØ+.5)/1ØØ
128Ø    PRINT
129Ø    PRINT "REOULAR OEPOSITS R
        EQUIREO:$";V(7)
13ØØ    OOTO 66Ø
131Ø    OOBUB 534Ø
132Ø    TITLE$=" FUTURE VALUE WIT
        H CABH FLOW8 ":OOSU8 TITL
        E8AR
133Ø    PRINT
134Ø    GOSU8 472Ø
135Ø    GOSU8 476Ø
136Ø    PRINT "CABH FLOW (+/-)"
137Ø    PRINT
138Ø    V(1)=Ø
139Ø    FOR I=1 TO V(4)
14ØØ    PRINT "CASH FLOW - YEAR $
        ";I
141Ø    INPUT A$
142Ø    A=VAL(A$)
143Ø    V(1)=V(1)+A*(1+V(3))^(V(4
        )-I)
144Ø    NEXT I
145Ø    V(1)=INT(V(1)*1ØØ+.5)/1ØØ

146Ø    OOSU8 497Ø
147Ø    TE=V(1)
148Ø    OO8U8 515Ø
149Ø    OOTO 66Ø
15ØØ    OOSUB 534Ø
```

```
1510  TITLE$=" WITHORAWAL OF FU
      NOS ";GOSUB TITLEBAR
1520  PRINT
1530  OOSUB 4630
1540  PRINT "*REGULAR WITHORAWA
      L $"
1550  C=7
1560  GOSUB 3850
1570  PRINT "*";
15B0  GOSUB 4720
1590  PRINT "*";
1600  GOSUB 4760
1610  IF E=4 THEN 1630
1620  GOSUB 4800
1630  GOSUB 4850
1640  IF E<>2 THEN 1670
1650  V(2)=INT(V(B)*V(6)/V(3)*(
      1-(1+V(3)/V(6))^(-V(6)*Y)
      )*100+.5)/100
1660  GOSUS 5000
1670  IF E<>3 THEN 1B10
16B0  V(3)=.99
1690  I=0
1700  R=INT(V(2)*V(3)/V(6)*(1/(
      (1+V(3)/V(6))^(V(6)*Y)-1)
      +1)*100+.5)/100
1710  TE=ABB(V(3)-I)/2
1720  I=V(3)
1730  IF ABS(R-V(B))/V(B)<.0000
      5 THEN 1790
1740  IF R<V(B) THEN 1770
1750  V(3)=V(3)-TE
1760  GOTO 1700
1770  V(3)=V(3)+TE
1780  GOTO 1700
1790  V(3)=INT(V(3)*10000+.5)/1
      0000
1800  GOSUB 5030
1B10  IF E<>4 THEN 1B40
1B20  V(4)=LOG(V(6)*V(B)/(V(6)*
      V(B)-V(3)*V(2)))/(V(6)*LO
      G(1+V(3)/V(6)))
1B30  GOSUB 5060
1B40  1F E<>B THEN 660
1B50  V(B)=INT(V(2)*V(3)/V(6)*(
      1/((1+V(3)/V(6))^(V(6)*Y)
      -1)+1)*100+.5)/100
1B60  PRINT
1B70  PRINT "REGULAR WITHORAWAL
      S:$";V(B)
1BB0  GOTO 660
1B90  GOSUB 5340
1900  PRINT "NET PRESENT VALUE:
      $"
1910  PRINT
1920  PRINT "INITIAL INVESTMENT
      "
1930  C=1
1940  GOSUB 3B50
1950  GOSUB 4720
1960  GOSUB 4760
1970  PRINT "CASH FLOW (+/-)"
19B0  PRINT
1990  NV=-V(2)
2000  FOR I=1 TO V(4)
2010  PRINT "CABH FLOW - YEAR #
       ";I
2020  INPUT A$
2030  A=VAL(A$)
2040  NV=NV+A/((V(3)+1)^I)
2050  NEXT I
2060  NV=INT(NV*100+.5)/100
2070  PRINT
20B0  PRINT "NET PRESENT VALUE:
      $";NV
2090  TE=NV
2100  GOSUB 5150
2110  GOTO 660
2120  GOSUB 5340
2130  TITLE$=" LOANS ";GOBUB TI
      TLEBAR
2140  GOTOXY 21,5:PRINT "1) REG
      ULAR LOAN PAYMENTS"
2150  OOTOXY 21,6:PRINT "2) REM
      AINING LOAN LIABILITY"
2160  GOTOXY 21,7:PRINT "3) FIN
      AL LOAN PAYMENT"
2170  GOTOXY 21,B:PRINT "4) SIN
      GLE PAYMENT LOAN"
2180  GOTOXY 21,9:PRINT "5) LOA
      N AMORTIZATION SCHEOULE"
2190  GOTOXY 21,10:PRINT "6) CA
      LCULATOR MODE"
2200  GOTOXY 21,11:PRINT "7) RE
      TURN TO MAIN MENU"
2210  GOTOXY 21,13:PRINT "YOUR
      CHOICE?";
2220  A=INP(2)-4B
2230  IF A<1 OR A>7 THEN 2220
2240  ON A GOTO 2270,2690,2B70,
      3030,3140,2250,190
2250  GOSUB 4060
2260  GOTO 190
2270  GOSUB 5340
22B0  TITLE$=" REGULAR LOAN PAY
      MENTB ";GOSUB TITLEBAR
2290  PRINT
2300  PRINT "*";
2310  GOBUB 4670
2320  PRINT "*";
2330  GOBUB 4890
2340  PRINT "*";
2350  GOSUB 4720
2360  PRINT "*";
2370  GOBUB 4760
23B0  IF E=4 THEN 2400
2390  GOBUB 4800
2400  GOSUB 4850
2410  IF E<>2 THEN 2460
2420  V(2)=INT(V(7)*V(6)/V(3)*(
      1-(1+V(3)/V(6))^(-V(6)*Y)
      )*100+.5)/100
2430  PRINT
2440  PRINT "AMT OF PRINCIPAL:$
      ";V(2)
2450  GOTO 2670
2460  IF E<>3 THEN 2600
2470  V(3)=.99
24B0  I=0
2490  P=INT(V(7)*V(6)/V(3)*(1-(
      (1+V(3)/V(6))^(-V(6)*Y)))
      *100+.5)/100
2500  TE=ABS(V(3)-I)/2
2510  I=V(3)
2520  IF ABS(P-V(2))/V(2) < .00
      005 THEN 2580
2530  IF P<V(2) THEN 2560
2540  V(3)=V(3)+TE
2550  GOTO 2490
2560  V(3)=V(3)-TE
2570  GOTO 2490
25B0  V(3)=INT(V(3)*10000+.5)/1
      0000
2590  GOBUB 5030
2600  IF E<>4 THEN 2630
2610  V(4)=-LOG(1-V(3)*V(2)/(V(
      6)*V(7)))/(V(6)*LOG(V(3)/
      V(6)+1))
2620  GOSUB 5060
2630  IF E<>7 THEN 2670
2640  V(7)=INT(V(3)*V(2)/(V(6)*
      (1-(V(3)/V(6)+1)^(-V(6)*Y
      )))*100+.5)/100
2650  PRINT
2660  PRINT "REQ PAYMENT:$";V(7
      )
2670  GOSUB 5210
26B0  GOTO 2120
2690  GOBUB 5340
2700  TITLE$=" REMAININO LOAN L
      IABILITY ";GOSUB TITLEBAR
2710  PRINT
2720  OOBUB 4670
2730  GOSUB 4B90
2740  GOSUB 4720
2750  GOSUB 4850
2760  PRINT "LAST PAYMENT # WAS
      :"
2770  INPUT A$
2780  A=VAL(A$)
2790  FOR J=1 TO A
2800  I=INT(P*V(3)/V(6)*100+.5)
      /100
2810  P=P+I-V(7)
2B20  NEXT J
2830  LI=INT(P*100+.5)/100
2840  PRINT
2850  PRINT "LIABILITY AFTER ";
      A;" PAYMENTS:$";LI
2860  GOTO 2670
2870  GOSUB 5340
28B0  TITLE$=" LAST LOAN PAYMEN
      T ";GOSUB TITLEBAR
2B90  PRINT
2900  GOSUB 4670
2910  GOSUB 4890
2920  GOBUB 4720
2930  OOSUB 4930
2940  GOSUB 4850
2950  FOR J=1 TO V(6)*Y
2960  I=INT(P*V(3)/V(6)*100+.5)
      /100
2970  P=P+I-V(7)
29B0  NEXT J
2990  LP=INT(P*100+.5)/100+V(7)

3000  PRINT
3010  PRINT "LAST PAYMENT:$";LP

3020  BOTO 2670
3030  GOSUB 5340
3040  TITLE$=" SINBLE PAYMENT L
      OAN ";GOSUB TITLEBAR
3050  PRINT
3060  GOSUB 4670
3070  GOSUB 4720
3080  GOSUB 4930
3090  GOSUB 4B50
3100  V(1)=INT(V(2)*(1+V(3)/V(6
      ))^(Y*V(6))*100+.5)/100
3110  PRINT
3120  PRINT "TOTAL OWEO:$";V(1)
3130  GOTO 2670
3140  C5=0
3150  N5=0
3160  F=0
3170  P1=0
31B0  I1=0
3190  GOSUB 5340
3200  TITLE$=" LOAN AMORTIZATIO
      N SCHEOULE ";GOSUB TITLEB
      AR
3210  GOSUB 4670
3220  OOSUB 4B90
3230  GOSUB 4720
3240  GOSUB 4930
3250  PRINT "# OF PAYMENTS YEAR
      LY"
3260  GOSUB 3B50
3270  PRINT "ENTER THE PERIOO O
      F THE YEAR IN WHICH THE L
      OAN BEGAN"
32B0  INPUT N
3290  NE=N
3300  NP=(V(4)*12+V(5))/(12/V(6
      ))
3310  NY=INT(((N-1)+NP)/V(6)+.9
      9)
3320  PRINT "ENTER THE RANGE OF
       YEARS YOU'O LIKE TO EXAM
      INE (FIRST, LAST)"
3330  INPUT F1,L1
3340  IF L1<=NY THEN 3360
3350  L1=NY
3360  FOR J1=1 TO L1
3370  IF J1<F1 THEN 3390
33B0  GOBUB 5250
3390  FOR J=1 TO V(6)-N+1
```

```
3400  I=INT(P*V(3)/V(6)*100+.5)
      /100
3410  N5=N5+1
3420  PP=V(7)-I
3430  IF J1<>NY THEN 3470
3440  IF N5<>NP THEN 3470
3450  PP=P
3460  F=1
3470  IF J1<F1 THEN 3500
3480  PRINT TAB(5);MID$(STR$(N5
      ),2,LEN(STR$(N5))-1);TAB(
      S1);INT(P*100+.5)/100;
3490  PRINT TAB(S2);INT(PP*100+
      .5)/100;Q$;TAB(S3);
3500  P=P+I-V(7)
3510  IF F=0 THEN 3540
3520  P=0
3530  J=V(6)
3540  IF J1<F1 THEN 3570
3550  PRINT 1;TAB(S4);INT(P*100
      +.5)/100;
3560  PRINT
3570  I1=I1+I
3580  P1=P1+PP
3590  C5=C5+1
3600  IF C5<>D5 THEN 3670
3610  IF J1<F1 THEN 3670
3620  GOSUB 5210
3630  GOSUB 5340
3640  C5=0
3650  IF J=V(6)-N+1 THEN 3670
3660  GOSUB 5250
3670  NEXT J
3680  IF J1<F1 THEN 3790
3690  IF F=0 THEN 3720
3700  GOTOXY 0,0
3710  PRINT "FINAL PAYMENT :$";
      INT((PP+I)*100+.5)/100
3720  PRINT
3730  PRINT "TOTAL INT PAID IN
      YR ";J1;":$";INT(I1*100+.
      5)/100
3740  PRINT "TOTAL PRINC PAID I
      N YR ";J1;":$";INT(P1*100
      +.5)/100
3750  IF F=1 THEN 3830
3760  IF J1=L1 THEN 3830
3770  GOSUB 5210
3780  GOSUB 5340
3790  C5=0
3800  P1=0
3810  I1=0
3820  N=1
3830  NEXT J1
3840  GOTO 2670
3850  C=C+1
3860  IF C<>3 THEN 3890
3870  PRINT V(3)*100,
3880  GOTO 3900
3890  PRINT V(C),
3900  INPUT A$
3910  IF LEN(A$)<>0 THEN 3930
3920  RETURN
3930  IF A$<>"MR" THEN 3990
3940  PRINT "MEM=";M;"  USE AS
      VARIABLE HERE (Y/N)"
3950  INPUT A$
3960  IF A$="N" THEN 3900
3970  V(C)=M
3980  RETURN
3990  IF A$="X" THEN E=C:RETURN

4000  IF A$="x" THEN E=C:RETURN

4010  V(C)=VAL(A$)
4020  IF C<>3 THEN 4040
4030  V(C)=V(C)/100
4040  RETURN
4050  REM CALCULATOR MODE
4060  GOSUB 5340:TITLE$=" Calcu
      lator Mode ":GOSUB TITLEB
      AR
4070  M5=0
4080  GOSUB 4410
4090  INPUT A$
4100  IF ASC(A$)>57 THEN 4130
4110  T=VAL(A$)
4120  GOTO 4090
4130  FOR I=1 TO 8
4140  IF A$<>MID$(V$,I,1) THEN
      4170
4150  PRINT V(I)
4160  T=V(I)
4170  NEXT I
4180  FOR J=1 TO 6
4190  IF A$<>MID$(C2$,(J-1)*2+1
      ,2) THEN 4210
4200  ON J GOSUB 4460,4480,4500
      ,4520,4540,4560
4210  NEXT J
4220  FOR K=1 TO 4
4230  IF A$<>MID$(C$,K,1) THEN
      4250
4240  ON K GOSUB 4290,4340,4410
      ,4440
4250  NEXT K
4260  IF M5=0 THEN 4090
4270  M5=0
4280  RETURN
4290  FOR I=1 TO 8
4300  PRINT MID$(V$,I,1);"  ";V
      (I)
4310  NEXT I
4320  PRINT
4330  RETURN
4340  PRINT "IN WHAT VARIABLE "
      ;
4350  INPUT A$
4360  FOR I=1 TO 8
4370  IF A$<>MID$(V$,I,1) THEN
      4390
4380  V(I)=M
4390  NEXT I
4400  RETURN
4410  COLOR 2,1:GOTOXY 0,0:PRIN
      T C0$;" ";C1$;" MEM=";M:C
      OLOR 1,1
4420  PRINT
4430  RETURN
4440  M5=1
4450  RETURN
4460  M=M+T
4470  GOTO 4570
4480  M=M-T
4490  GOTO 4570
4500  M=M*T
4510  GOTO 4570
4520  M=M/T
4530  GOTO 4570
4540  T=M
4550  GOTO 4570
4560  M=0
4570  PRINT "MEM=";M
4580  RETURN
4590  PRINT "*FUTURE VALUE $"
4600  C=0
4610  GOSUB 3850
4620  RETURN
4630  PRINT "*PRESENT VALUE $"
4640  C=1
4650  GOSUB 3850
4660  RETURN
4670  PRINT "PRINCIPAL $"
4680  C=1
4690  GOSUB 3850
4700  P=V(C)
4710  RETURN
4720  PRINT "ANNUAL INT RATE (%
      )"
4730  C=2
4740  GOSUB 3850
4750  RETURN
4760  PRINT "FOR # OF YEARS"
4770  C=3
4780  GOSUB 3850
4790  RETURN
4800  PRINT "FOR # OF MONTHS"
4810  C=4
4820  GOSUB 3850
4830  Y=V(C-1)+V(C)/12
4840  RETURN
4850  PRINT "# OF PERIODS (COMP
      OUNDING, DEPOSITS, WITHDR
      AWALS, PAYMENTS) YEARLY"
4860  C=5
4870  GOSUB 3850
4880  RETURN
4890  PRINT "PAYMENTS $"
4900  C=6
4910  GOSUB 3850
4920  RETURN
4930  PRINT "TERM OF LOAN:"
4940  GOSUB 4760
4950  GOSUB 4800
4960  RETURN
4970  PRINT
4980  PRINT "FUTURE VALUE:$";V(
      1)
4990  RETURN
5000  PRINT
5010  PRINT "REQUIRED INVESTMEN
      T:$";V(2)
5020  RETURN
5030  PRINT
5040  PRINT "ANNUAL INT RATE (%
      ) REQUIRED:";V(3)*100
5050  RETURN
5060  V(5)=V(4)-INT(V(4))
5070  V(5)=INT(INT(12*V(5)*10+.
      5)/10)
5080  V(4)=INT(V(4))
5090  IF V(5)<>12 THEN 5120
5100  V(4)=V(4)+1
5110  V(5)=0
5120  PRINT
5130  PRINT "# OF YEARS AND MON
      THS:";V(4);",";V(5)
5140  RETURN
5150  PRINT
5160  IF TE>=0 THEN 5190
5170  PRINT "THIS IS A LOSING I
      NVESTMENT."
5180  RETURN
5190  PRINT "THIS IS A PROFITAB
      LE INVESTMENT."
5200  RETURN
5210  PRINT
5220  COLOR 2,2:PRINT "Press an
      y key to continue";:COLOR
       1,1
5230  A = INP(2)
5240  RETURN
5250  GOSUB 5340
5260  PRINT "LOAN AMORTIZATION
      SCHEDULE FOR YR ";J1
5270  PRINT "PRIN $";V(2);"  RA
      TE ";V(3)*100;"%";"  PAYM
       $";V(7)
5280  PRINT
5290  COLOR 3,1
5300  PRINT TAB(5);"#";TAB(11);
      "BEG BAL";TAB(26);"PRINC"
      ;TAB(41);"INT";
5310  PRINT TAB(56);"END BAL"
5320  COLOR 1,1
5330  RETURN
5340  CLEARW 2:FULLW 2:GOTOXY 0
      ,0
5350  RETURN
5360  TITLEBAR:
5370  A# = GB : GINTIN = PEEK(A
      #+8)
5380  POKE GINTIN+0,PEEK(SYSTAB
      +8) : POKE GINTIN+2,2
5390  S# = GINTIN+4 : TITLE$ =
      TITLE$ + CHR$(0)
5400  POKE S#,VARPTR(TITLE$) :
      GEMSYS(105)
5410  RETURN
```

©

Fast IBM Batch File Editor

Tony Roberts, Production Director

Now it's quick and easy to edit and fine-tune batch files with this DOS utility. It works on any IBM PC or PCjr with an 80-column monitor.

The power of the batch file quickly becomes evident to anyone who works regularly in PC-DOS. The hardy AUTOEXEC.BAT handles a variety of chores each time the system is booted, and any number of other .BAT files stand by, ready to help with such tasks as initializing applications, sending out printer codes, and presenting program menus.

The problem with batch files is that to be effective and helpful, they need to be adjusted as your system grows and your applications change. Performing the necessary batch-file maintenance, however, is often so cumbersome that it's discouraging. Loading a full-blown word processor to edit a five- to ten-line batch file can be a lot more time and trouble than it's worth.

"EdBat" solves this problem by focusing all its energy on your batch files. EdBat is without frills, but it's fast and easy to use.

What EdBat Does

EdBat is a full-screen editor with very limited features. Because it is designed for speed, it limits itself to files of fewer than 512 bytes—adequate for most batch files. (If your file is longer, you're probably better off with a more sophisticated editor.)

When called, the program clears the screen and displays the file you want to edit. Using the cursor keys, you can move to the appropriate place, make the necessary changes, and press Alt-S to save the edited file. It is not impossible to open a file, edit it, close it, and be back at the DOS prompt in as little as 15 seconds.

The price you pay for this fast operation is that EdBat has very few features. You're essentially limited to the regular character keys and the cursor keys. The Insert key does not work, the Delete key does not work, nor do the function keys perform any function. The Backspace key moves the cursor back a character, but it does not perform a delete.

If you were writing a novel, these restrictions would be serious, but in batch file editing, none of them is particularly restrictive. With batch files, you're usually just performing one or two simple operations such as adding, deleting, or correcting a line. EdBat can handle all these tasks efficiently.

Using The Program

EdBat is a machine language program that is activated from the DOS prompt. The program listed below, "EdBat Loader," is a BASIC program that creates the file EDBAT.COM from the information in BASIC DATA statements. Type in EdBat Loader using the "IBM Automatic Proofreader," save a copy to disk, and then run it once to create EDBAT.COM.

To run EdBat, enter this line from the DOS prompt:

```
EDBAT filename
```

(The EDBAT.COM file must be on the disk in the current drive when you enter this command.) *Filename* is the name of the file you wish to edit. Full drive and subdirectory specifications are allowed when indicating a filename. If the file is too long or if EdBat is unable to open the file, the program will print a message and exit. If the file you have specified does not exist, EdBat assumes you are creating a new file.

In a matter of seconds, the file you are to edit is displayed on the screen below a line containing the program title and the name of the current file. If you have started a new file, the screen's work area will be blank.

Use the cursor keys to move around the file, editing as needed. Notice that a triangle signals the end of each line. If you decide to cut a line short, move to the appropriate spot and press Enter. A triangle is inserted and the cursor moves to the beginning of the next line. The screen may continue to show characters beyond the end-of-line marker, but they will be ignored when the file is saved.

To delete an entire line, simply move to the first position on that line and press Enter. An end-of-line marker appears at that spot, indicating that the line will be ignored.

Inserting a line is slightly more difficult since there is no insert function. Move the cursor to the end-of-line marker on the line that will precede your new line. Press Ctrl-Y and a down-arrow character (↓) will replace the end-of-line marker. Add the new line right after the down arrow and press Enter as usual. When the file is saved, the lines will be adjusted.

Saving The Changes

When you're finished editing, press

Alt-S to save the file. The program's save routine reads the screen and saves what it sees to your file. It begins with the first line of the text area and continues until it finds a space in the first position of any line. EdBat ignores any characters in a line which follow the first end-of-line marker.

The only other option the program offers is Alt-Q, the Quit option, which returns you to DOS without changing the original file. In nearly every case, your entire file will fit easily on the screen. If part of your file scrolls off the screen, use Alt-Q to quit and find another method of editing the file. EdBat cannot save what it cannot see.

Unlike many word processors, EdBat does not make a backup of your original file. In most cases, though, a backup of a very short file is superfluous. For years, EDLIN, the line editor included with PC-DOS, had been my batch file editor. Eventually, though, I lost patience with it over the time it spent writing backup files and went to work on EdBat.

EdBat Command Summary

Alt-Q	Quit
Alt-S	Save
Ctrl-Y	Multistatement delimiter (prints as a down arrow)
Enter	End-of-line (prints as left-pointing triangle)
Space	Space in first position of line signals text end

EdBat Loader

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

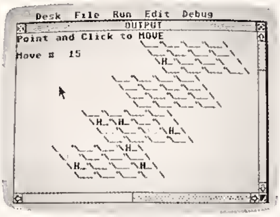
```
HL 10 CLS
EI 20 DPEN "EDBAT.CDM" AS 1 LEN
      = 1
FC 30 FIELD 1, 1 AS A$
GP 40 PRINT:PRINT"Writing EDBAT.
      CDM to disk. Please wait."
GC 50 FDR I=1 TO B:READ B$:GDSUB
       130:NEXT I
CN 60 FOR I=1 TD 75:B$="5F":GOSU
      B 130:NEXT I
GM 70 B$="24":GDSUB 130
OF B0 FDR I=1 TD 74:B$="0":GDSUB
       130:NEXT I
HG 90 FDR I=1 TD 64B:READ 8$: GD
      SUB 130: NEXT I
OG 100 CLOSE
MA 110 PRINT:PRINT"ED8AT.CDM has
        been created."
LO 120 END
MF 130 REM write byte to disk
OD 140 LSET A$ = CHR$(VAL("&H"+B
       $))
LK 150 PUT #1
MH 160 RETURN
DN 170 DATA E9, E0, 0, 45, 64, 4
       2, 61, 74, 2, 0
CE 180 DATA 2, 50, 6C, 65, 61, 7
       3, 65, 20, 73, 70, 65, 63
       , 69, 66, 79, 20
EK 190 DATA 66, 69, 6C, 65, 6E,
       61, 6D, 65, 2E, D, A, 24,
        45, 72, 72, 6F
EN 200 DATA 72, 20, 6F, 70, 65,
       6E, 69, 6E, 67, 20, 66, 6
       9, 6C, 65, 2E, D
OL 210 DATA A, 24, 46, 69, 6C, 6
       5, 20, 74, 6F, 6F, 20, 6C
       , 6F, 6E, 67, 2E
GP 220 DATA D, A, 24, FC, BF, 54
       , 1, BE, B0, 0, AC, A2, 9
       9, 1, FE, E
JJ 230 DATA 99, 1, 3C, 0, 75, 9,
        BA, A1, 1, EB, 7, 2, EB,
        25, 2, AC
FH 240 DATA 3C, D, 74, 3, AA, EB
       , F8, EB, BA, 1, 73, E, 3
       D, 2, 0, 74
BA 250 DATA 30, BA, BC, 1, EB, E
       C, 1, EB, A, 2, BA, 26, 4
       , BB, 1E, 9A
QI 260 DATA 1, BB, E, 9F, 1, B4,
        3F, CD, 21, 38, 6, 9F, 1
       , 75, C, EB
AD 270 DATA A0, 1, BA, D2, 1, EB
       , C8, 1, E8, E9, 1, A3, 9
       6, 1, E8, 91
HA 2B0 DATA 1, EB, CB, 1, BA, 3,
        1, E8, B9, 1, C6, 6, 9E,
        1, 0, C6
KF 290 DATA 6, 9D, 1, 14, EB, 60
       , 1, BE, 55, 1, 33, C9, 8
       A, E, 99, 1
HP 300 DATA AC, BA, D0, B0, FA,
       61, 72, 3, 80, E2, DF, E8
       , 9C, 1, E2, F0
HL 310 DATA C6, 6, 9E, 1, 2, C6,
        6, 9D, 1, 0, E8, 3A, 1,
       B3, 3E, 96
OJ 320 DATA 1, 0, 74, 1F, FC, 8E
       , 26, 4, 8B, E, 96, 1, AC
       , 8A, D0, 80
JN 330 DATA FA, D, 75, 7, B2, 11
       , EB, 71, 1, 82, D, E8, 6
       C, 1, E2, EC
HM 340 DATA E8, 14, 1, B4, 0, CD
       , 16, 3C, 0, 74, 13, 3C,
       D, 74, A, 3C
MO 350 DATA B, B4, 4B, 74, 1C, 3
       C, 19, 72, EA, E8, 1F, 1,
        E8, E5, B0, FC
PE 360 DATA 4B, 75, E, B0, 3E, 9
       E, 1, 2, 74, D9, FE, E, 9
       E, 1, E8, E6
OB 370 DATA 0, B0, FC, 48, 75, E
       , 80, 3E, 9D, 1, 0, 74, C
       6, FE, E, 9D
DB 3B0 DATA 1, EB, D3, 0, B0, FC
       , 4D, 75, E, 80, 3E, 9D,
       1, 4F, 74, 83
OJ 390 DATA FE, 6, 9D, 1, EB, C0
       , 0, 80, FC, 50, 75, E, B
       0, 3E, 9E, 1
NJ 400 DATA 1B, 74, A0, FE, 6, 9
       E, 1, E8, AD, 0, B0, FC,
       10, 75, 6, EB
OK 410 DATA FD, 0, EB, F, 1, B0,
        FC, 1F, 75, 89, C7, 6, 9
       6, 1, 0, 0
JO 420 DATA BF, 26, 4, C6, 6, 9E
       , 1, 2, C6, 6, 9D, 1, 0,
       EB, B7, 0
ML 430 DATA C6, 6, 95, 1, 0, B4,
        B, CD, 10, 3C, 20, 74, 4
       E, 3C, 11, 75
GG 440 DATA 9, FE, 6, 9E, 1, EB,
        6F, 0, EB, DE, B0, 3E, 9
       5, 1, 50, 77
OJ 450 DATA 1B, B4, 8, CD, 10, 3
       C, 11, 74, 10, 3C, 19, 75
       , 1C, B0, D, AA
OE 460 DATA FF, 6, 96, 1, B0, A,
        EB, 11, 90, 80, D, B4, A
       , AB, B3, 6
AD 470 DATA 96, 1, 2, FE, 6, 9E,
        1, EB, AF, AA, FF, 6, 96
       , 1, FE, 6
LC 480 DATA 95, 1, FE, 6, 9D, 1,
        E8, 2E, 0, E8, BF, FB, B
       A, 55, 1, 84
CF 490 DATA 3C, B9, 0, 0, CD, 21
       , 73, 9, BA, 8C, 1, E8, 6
       5, 0, EB, B3
LA 500 DATA 0, BB, DB, BB, E, 96
       , 1, 8A, 26, 4, B4, 40, C
       D, 21, E8, 21
HO 510 DATA 0, EB, 5B, 0, EB, 6D
       , 0, BA, 36, 9E, 1, 8A, 1
       6, 9D, 1, B4
CG 520 DATA 2, CD, 10, C3, FB, B
       A, 55, 1, B0, 2, B4, 3D,
       CD, 21, A3, 9A
LC 530 DATA 1, C3, BB, 1E, 9A, 1
       , 84, 3E, CD, 21, C3, 8A,
        D0, 80, FA, D
PH 540 DATA 74, B, EB, 25, 0, FE
       , 6, 9D, 1, C3, 82, 11, E
       B, 18, 0, B2
JO 550 DATA D, EB, 16, 0, B2, A,
        E8, 11, 0, FE, 6, 9E, 1,
        C6, 6, 9D
IO 560 DATA 1, 0, C3, 50, B4, 9,
        CD, 21, 5B, C3, 84, 2, C
       D, 21, C3, B4
IB 570 DATA F, CD, 10, BB, 3E, 9
       C, 1, B4, 0, B0, 2, CD, 1
       0, B4, 5, B0
BF 5B0 DATA 0, CD, 10, C3, CD, 2
       0
```

©

3-D Tic-Tac-Toe For Atari ST

David Bohlke

This new rendition of an old favorite lets you match wits against the ST computer in a three-dimensional contest. You can even, if you like, make changes to the program which will make the computer play more aggressively or more cautiously. "3-D Tic-Tac-Toe" runs on any Atari 520ST or 1040ST computer with a color monitor.



Point and Click to MOVE

Move # 15

"3-D Tic-Tac-Toe For Atari ST" challenges you to best the computer in a three-dimensional strategic simulation.

"3-D Tic-Tac-Toe" is a strategy game where you take on the Atari ST in a battle of wits. The object of the game is similar to the traditional Tic-Tac-Toe game, except this version takes place in a simulated three-dimensional space containing four game boards. To win, you must place four pieces in a row. The row may extend across a single plane or vertically though all four planes. Though it's not a flawless player, the ST will provide you with a formidable opponent.

Entering Tic-Tac-Toe

Type in the program as listed and save it to disk. The program works in either low- or medium-resolution modes. When you run the program, it randomly selects whether you or the computer should go first. The computer needs only a few seconds to pick its move and places a red uppercase C at the selected square. (The ST takes less time to move if you refrain from moving the mouse pointer around while it is calculating; moving the pointer freezes normal BASIC operations. In addition, you should avoid moving the slider bars on the output window, since this may jumble part of the game board.)

It's your turn when the screen prompt appears. Use the mouse to move to the square of your choice, then click the left mouse button. Due to the slowness of ST BASIC, you may need to hold the button down for as long as one second before the computer recognizes your choice. A blue uppercase *H* appears on the square you have chosen. The *H*, of course, stands for the Human, you, and the *C* stands for Computer.

Programmed Strategy

You may be interested in learning how the ST plays this simple strategy game. The computer does not use a "look-ahead" technique, but rather determines its move by assigning a numeric value to each empty square. This value is explained in the table, which shows a sample Tic-Tac-Toe combination of four squares in a row, along with the corresponding BASIC line number that assigns the value.

Combination Values

Line	Pattern	Value
540	HHHH	human wins
540	CCCC	computer wins
550	H_HH	33 points
560	_H_H	5 points
570	__H_	2 points
580	CC_C	77 points
590	C__C	6 points
600	_C__	1 point

Each computer piece is stored with a value of 5 in the V( ) array, and each human piece has a value of 1 in the array. So if a row of four squares contains two computer pieces, that combination has a value of 10. Lines 540–600 then convert these combination values into point values, which are evaluated to choose the next move. Note that the order of pieces in the table has no significance: What matters is the number of pieces and blanks. In the third entry, for instance, the sequence H_HH merely indicates that the row contains one blank and three human pieces, in any order. No value is assigned to a row that contains both computer pieces and human pieces since it's clearly impossible to win on that row.

This game is designed so that the computer plays a nearly equal balance of offense and defense. If you would like the computer to play more aggressively, increase the values for offensive moves in lines 590 and 600. For a more conservative game, you can increase the values in lines 560 and 570. With a little experience, you'll find that a change of just one or two points in these four lines will make a significant difference in the computer's move strategy.

3-D Tic-Tac-Toe

```
1ØØ   fullw 2:clearw 2
11Ø   dim b(64),v(64),x(64),m(6
      4,28):gosub 67Ø
12Ø   ' new game
13Ø   clearw 2:color 1:print:fo
      r s=1 to 64:gosub 87Ø
14Ø   gotoxy x-1,y:print"\__\";
      :next
15Ø   for i=1 to 64:b(i)=Ø:x(i)
      =Ø:v(i)=Ø:next:w(1)=Ø:mv=
      Ø
16Ø   randomize Ø:if rnd(1)<.5
      then s=int(rnd(1)*64)+1:g
      osub 84Ø:color 2:goto 37Ø
17Ø   ' human moves
18Ø   gosub 84Ø:color 4:print:g
      otoxy Ø,Ø:print"Point and
       Click to MOVE"
19Ø   gosub mousexy:mx=int(msx/
      9):my=int(msy/9.3)
2ØØ   sq=Ø:if msb<>1 then 19Ø
21Ø   for s=1 to 64:gosub 87Ø
22Ø   if y=my-2 and abs(x-mx)<=
      1 then sq=s
23Ø   next:if sq=Ø then 19Ø
24Ø   s=sq:gosub 87Ø
25Ø   if b(s)<>Ø then 19Ø
26Ø   sx=1:gotoxy x,y:print"H_"
      ;:b(s)=1:v(s)=Ø:gosub 52Ø
27Ø   if w(1)>Ø then 44Ø
28Ø   ' computer moves
29Ø   gosub 84Ø:color 2:print:g
      otoxy Ø,Ø:print"Atari ST'
      s Move          "
3ØØ   sx=Ø:for s=1 to 64:if b(s
      )>Ø or x(s)=Ø then 31Ø el
      se v(s)=Ø:gosub 52Ø
31Ø   next
32Ø   s=Ø:h=Ø:for i=1 to 64
33Ø   if v(i)=h and rnd(1)<.3 a
      nd h>Ø then h=v(i):s=i
34Ø   if v(i)>h then h=v(i):s=i
35Ø   next
36Ø   if s=Ø then gotoxy Ø,Ø:pr
      int"    DRAW game          "
      ;:a$="D":color 1:w(1)=1:
       w(2)=2:w(3)=3:w(4)=4:got
      o 46Ø
37Ø   gosub 87Ø:b(s)=5:v(s)=Ø
38Ø   for i=1 to 4:gotoxy x,y:p
      rint" *";:sound 1,8,1,4,1
      Ø
39Ø   gotoxy x,y:print"C_";:sou
      nd 1,8,1,5,1Ø:next:sound
      1,Ø,Ø,Ø,Ø
4ØØ   sx=1:for i=1 to 64:x(i)=Ø
      :next:gosub 52Ø
41Ø   if w(1)>Ø then 45Ø
42Ø   goto 17Ø
43Ø   ' game over
44Ø   gotoxy Ø,Ø:print"You WIN
                            ";:a
      $="H":goto 46Ø
45Ø   gotoxy Ø,Ø:print"Computer
       WINS    ";:a$="C"
46Ø   gotoxy Ø,1:print"CLICK fo
      r new game";
47Ø   for i=1 to 4:s=w(i):gosub
       87Ø:gotoxy x,y:print a$;
      :next:for i=1 to 99:next
48Ø   sound 1,8,5,5,1Ø:sound 1,
      Ø,Ø,Ø,Ø
49Ø   for i=1 to 4:s=w(i):gosub
       87Ø:gotoxy x,y:print"  "
      ;:next:for i=1 to 99:next
5ØØ   gosub mousexy:if msb<>Ø t
      hen 12Ø else 47Ø
51Ø   ' adjust value array V(64
      ) for computer move at sq
      uare s
52Ø   eg=Ø:j=1:for i=1 to m(s,Ø
      )
53Ø   p=Ø:for k=1 to 4:p=p+b(m(
      s,j)):j=j+1:next:q=Ø
54Ø   if p=4 or p=2Ø then for k
      =Ø to 3:w(k+1)=m(s,j+k-4)
      :next
55Ø   if p=3 then q=33:goto 62Ø
56Ø   if p=2 then q=5:goto 62Ø
57Ø   if p=1 then q=2:goto 62Ø
58Ø   if p=15 then q=77:goto 62
      Ø
59Ø   if p=1Ø then q=6:goto 62Ø
6ØØ   if p=5 then q=1:goto 62Ø
61Ø   if sx=1 then 62Ø else 66Ø
62Ø   v(s)=v(s)+q:if b(s)>Ø the
      n v(s)=Ø
63Ø   if sx=Ø then 66Ø
64Ø   for k=Ø to 3:if b(m(s,j+k
      -4))=Ø then x(m(s,j+k-4))
      =1
65Ø   next
66Ø   next:return
67Ø   ' load legal win combos i
      nto M(64,28)
68Ø   clearw 2:color 1:print" L
      oading DATA ..."
69Ø   for i=1 to 64:m(i,Ø)=Ø:ne
      xt
7ØØ   for i=1 to 16:a=i*4-3:for
       j=1 to 4:w(j)=a:a=a+1:ne
      xt:gosub 82Ø:next
71Ø   for i=1 to 4:for j=1 to i
      +48 step 16:n=j
72Ø   for k=1 to 4:w(k)=n:n=n+4
      :next:gosub 82Ø:next:next
73Ø   for i=1 to 16:for j=Ø to
      3:w(j+1)=j*16+i:next:gosu
      b 82Ø:next
74Ø   for i=1 to 28:for j=1 to
      4:read a:w(j)=a:next:gosu
      b 82Ø:next:return
75Ø   data 1,21,41,61,2,22,42,6
      2,3,23,43,63,4,24,44,64
76Ø   data 1,18,35,52,5,22,39,5
      6,9,26,43,6Ø,13,3Ø,47,64
77Ø   data 4,19,34,49,8,23,38,5
      3,12,27,42,57,16,31,46,61
78Ø   data 13,25,37,49,14,26,38
      ,5Ø,15,27,39,51,16,28,4Ø,
      52
79Ø   data 1,6,11,16,17,22,27,3
      2,33,38,43,48,49,54,59,64
8ØØ   data 4,7,1Ø,13,2Ø,23,26,2
      9,36,39,42,45,52,55,58,61
81Ø   data 1,22,43,64,4,23,42,6
      1,13,26,39,52,16,27,38,49
82Ø   for k=1 to 4:i=m(w(k),Ø)*
      4+1:m(w(k),Ø)=m(w(k),Ø)+1
83Ø   for p=1 to 4:m(w(k),i)=w(
      p):i=i+1:next:next:return
84Ø   color 1:mv=mv+1:gosub clr
      prt:gotoxy Ø,2:print"Move
       # ";mv;:return
85Ø   clrprt:gotoxy Ø,Ø:print s
      pc(23);:return
86Ø   ' input s=square to move
      to, returns x,y as print
      position
87Ø   a=int((s-1)/16):y=a*4+3:b
      =s-a*16
88Ø   c=int((b-1)/4):y=y+c-2:x=
      (4-a)*4+c
89Ø   x=x+(b-c*4)*3-1:return
9ØØ   mousexy:poke contrl,124:p
      oke contrl+2,Ø
91Ø   poke contrl+6,Ø:vdisys(Ø)
92Ø   msx=peek(ptsout):msy=peek
      (ptsout+2):msb=peek(intou
      t):return
```

©

Rapid Transfer

Buck Childress

The Commodore 64's BASIC has no built-in search-and-replace function, so renaming variables in a program can be a very time-consuming job. With this utility, you can easily rename any type of variable in a BASIC program. Though it's written in machine language for extra speed, no machine language knowledge is needed to use it.

No matter how well you plan ahead, nearly every BASIC programmer needs to modify his or her work from time to time. Renaming variables is one of the most tedious and exacting tasks you will face as a BASIC programmer. You must painstakingly comb every line of the program to insure that you have changed every reference to the variable involved. Should one reference be overlooked, the program will refuse to run correctly, if at all. The longer the program, the more tiresome the task becomes, and the greater the risk of introducing errors. The next time you find yourself in this situation, give "Rapid Transfer" a try. It automatically renames any variable you choose, whether string, numeric, integer, or array. It's easy to use, and gets the job done in a jiffy.

Getting Started

Type in the program as listed, then save a copy to disk or tape. To install Rapid Transfer, simply type RUN and press RETURN. The program automatically loads a machine language routine into the memory area beginning at location 50000. Since this memory zone isn't part of BASIC program space, you can load and save BASIC programs without interference.

Next, load the BASIC program you want to work on. To activate Rapid Transfer, type SYS 50000 and press RETURN. It begins by asking you for the old variable name—the name of an existing variable which you want to change. Type in this name, then press RETURN. At this point, you're asked to supply a new name for the variable. Should you happen to make a mistake while answering a prompt, press the INST/DEL key (pressing it twice will start you at the beginning).

You can enter up to ten characters for each variable name, in case you like to use extended names such as HOUSE$ or MATH%. If the variable you want to change is an integer or string, you will not be able to enter any additional characters after pressing the % or $ key (BASIC syntax doesn't allow it). Also, you can enter a number only after you've entered a letter (another BASIC syntax rule). Should you enter different types of variables, such as renaming a numeric variable with a string variable, Rapid Transfer displays the message TYPE MISMATCH. You'll then be given the option of going ahead with the transfer or starting over.

If the variable you want to change is an array, press the asterisk (*) key. You can do this at any time while you are entering the variable names, and it has to be done only once. Note that Rapid Transfer can tell when a variable is an array and responds accordingly. It is not necessary to enter the parentheses which ordinarily indicate an array—just enter the name itself. For example, to enter an array that you DIMension as A(20), you would enter A, not A( ).

After you press the asterisk key, the message ARRAY? begins flashing at the top of your screen. This is your prompt to enter the number of dimensions in the array. Enter 1, 2, or 3, depending on whether the array has one, two, or three dimensions. After you answer the prompt, the message stops flashing. If you make a mistake or want to cancel the array option, press the English pound (£) key. Rapid Transfer will not change an array variable to a nonarray variable, or vice versa, nor will it change the number of dimensions in an array.

After entering the new variable name and pressing RETURN, you'll see the message ARE YOU SURE? (Y/N). Press Y to proceed or N if you wish to reenter your choices.

Prescan For Name Conflicts

The first thing you'll notice when Rapid Transfer begins working is the line numbers of your program flashing at the top of the screen. Rapid Transfer is prescanning every line of the program to see whether it already contains a variable with the new name that you have chosen. If a name conflict is found, Rapid Transfer displays a warning message. If the variable is an array, an asterisk appears next to its name (a two-dimensional array has two asterisks, and so forth).

After it finishes the prescan, Rapid Transfer displays the prompt ARE YOU SURE? (Y/N). If no name conflicts appeared, or if you wish to proceed despite the conflict, press Y. Press N if a conflict is found or if you simply change your mind.

Rapid Transfer now displays the lines of your program as it seeks out the old variables and renames them. If the old variable doesn't exist in your program, Rapid Transfer displays a warning message indicating that the designated variable can't be found. Again, array variable names are displayed with one, two, or three asterisks, depending on the number of dimensions in the array. When it's done, the program lets you continue with another change (press Y) or quit (press N).

Safety Features

Rapid Transfer has several built-in safety features to insure accurate operation. It won't change anything enclosed within quotation marks or anything which appears on a line following a REM or DATA statement. While scanning each line, it also checks for excessive length. If, for example, you decide to change the variable CO$ to COST$ and, as a result, one of the program lines will exceed the 80-character logical length, Rapid Transfer aborts operation and displays the line number where the excessive length occurred. It also displays that line as it currently appears in the program so that you can make any necessary adjustments.

In addition, Rapid Transfer can tell the difference between different kinds of variables. For example, let's say that you want to rename the numeric variable A to A1. Rapid Transfer will rename only the numeric variable A. It will not rename any integer, string, or array variables of the same name, nor will it inadvertently change a variable which happens to begin with *A*, such as AB. The same holds true for the other types of variables, including arrays. If you have a one-dimensional array named A, Rapid Transfer will not change a two- or three-dimensional array of the same name, or vice versa.

Rapid Transfer works equally well with extended variable names. If you have used HOUSE$ in a home budget program, Rapid Transfer will recognize it as HO$, exactly as the 64 does. The entire name is present in the program line, but only the first two characters are significant. So you can use and change extended variable names as much as you like, with variables of any type.

Rapid Transfer can be brought to a halt at any time by pressing the RUN/STOP key. Enter SYS 50000 to reactivate it.

Rapid Transfer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
GD 10  PRINTCHR$(147)CHR$(5)"LO
       ADING AND CHECKING DATA
       {SPACE}LINE:":J=50000:L=
       45:C=11
GS 20  PRINTCHR$(19)TAB(31)L:PR
       INT
CG 30  FORB=ØTOC:READA:POKEJ+B,
       A:X=X+A:NEXTB:READA
DR 40  IFX<>ATHENPRINT"ERROR IN
        DATA LINE:"L:END
SJ 50  X=Ø:J=J+12:L=L+5:IFL<685
       THEN2Ø
GQ 60  IFL=685THENC=9:GOTO2Ø
EE 70  PRINT"DATA OK AND LOADED
       ...":PRINT:PRINT"SYS 500
       ØØ TO ACTIVATE...":END
HM 80  DATA32,59,2ØØ,133,198,13
       3,253,162,96,134,251,142
       ,1793
GF 90  DATA138,2,157,Ø,2Ø1,232,
       2Ø8,25Ø,2Ø2,142,224,2Ø1,
       1957
PG 100  DATA142,225,2Ø1,169,94,
        133,252,141,247,2Ø1,162
        ,1Ø,1977
HH 110  DATA32,71,2ØØ,133,254,1
        66,252,169,1ØØ,157,Ø,4,
        1538
JD 120  DATA173,134,2,157,Ø,216
        ,32,162,2ØØ,173,141,2,1
        392
AC 130  DATA2Ø1,2,176,246,32,22
        8,255,2Ø1,13,2Ø8,3,76,1
        641
DA 140  DATA179,196,2Ø1,2Ø,2Ø8,
        3,76,136,196,2Ø1,92,24Ø
        ,1748
AS 150  DATA51,2Ø1,42,2Ø8,87,14
        1,239,2Ø1,173,33,2Ø8,14
        1,1725
GX 160  DATA25,216,32,36,2ØØ,2Ø
        6,221,2Ø1,2Ø8,17,32,36,
        143Ø
ER 170  DATA2ØØ,238,222,2Ø1,48,
        6,32,116,2ØØ,76,2Ø3,195
        ,1737
DJ 1B0  DATA32,11Ø,2ØØ,2Ø6,24B,
        2Ø1,32,162,2ØØ,32,228,2
        55,19Ø6
QC 190  DATA2Ø1,92,2Ø8,11,169,Ø
        ,141,239,2Ø1,32,11Ø,2ØØ
        ,16Ø4
JD 200  DATA76,247,195,2Ø1,49,1
        44,2Ø6,2Ø1,52,176,2Ø2,1
        41,189Ø
JP 210  DATA25,4,56,233,49,141,
        234,2Ø1,32,116,2ØØ,141,
        1432
EK 220  DATA25,216,14Ø,222,2Ø1,
        76,134,195,2Ø1,36,24Ø,4
        ,169Ø
EB 230  DATA2Ø1,37,2Ø8,31,166,2
        51,48,14,174,96,2Ø1,24Ø
        ,1667
PA 240  DATA236,141,253,2Ø1,141
        ,254,2Ø1,76,34,196,174,
        192,2Ø99
FR 250  DATA2Ø1,24Ø,222,141,255
        ,2Ø1,133,254,76,9Ø,196,
        166,2175
PC 260  DATA254,224,1Ø,144,13,2
        4Ø,2,176,2Ø4,162,88,32,
        1549
EM 270  DATA71,2ØØ,23Ø,254,2Ø8,
        195,2Ø1,48,144,191,2Ø1,
        58,2ØØ1
BF 280  DATA176,16,174,96,2Ø1,1
        64,251,16,3,174,192,2Ø1
        ,1664
KA 290  DATA224,Ø,24Ø,173,2Ø8,8
        ,2Ø1,65,144,167,2Ø1,91,
        1722
MA 300  DATA176,163,23Ø,254,166
        ,251,157,Ø,2Ø1,23Ø,251,
        48,2127
FM 310  DATA12,174,236,2Ø1,224,
        2,176,15,238,236,2Ø1,2Ø
        8,1923
XS 320  DATA1Ø,174,237,2Ø1,224,
        2,176,3,238,237,2Ø1,32,
        1735
BM 330  DATA21Ø,255,23Ø,252,165
        ,251,141,247,2Ø1,76,121
        ,195,2344
XF 340  DATA174,247,2Ø1,16,112,
        166,211,32,21Ø,255,2Ø2,
        2Ø8,2Ø34
DA 350  DATA25Ø,142,192,2Ø1,142
        ,237,2Ø1,142,247,2Ø1,14
        2,255,2352
CR 360  DATA2Ø1,169,145,32,53,2
        ØØ,169,192,162,27,16Ø,1
        74,1684
JM 370  DATA133,251,132,252,76,
        116,195,166,252,173,33,
        2Ø8,1987
GC 380  DATA157,Ø,216,173,96,2Ø
        1,24Ø,197,166,251,16,22
        6,1939
CD 390  DATA173,192,2Ø1,24Ø,188
        ,141,2Ø8,2Ø1,173,193,2Ø
        1,141,2252
RF 400  DATA2Ø9,2Ø1,169,Ø,141,1
        38,2,141,98,2Ø1,157,Ø,1
        457
AA 410  DATA2Ø1,173,254,2Ø1,2Ø5
        ,255,2Ø1,24Ø,5,162,44,3
        2,1973
RC 420  DATA71,2ØØ,162,64,32,71
        ,2ØØ,32,162,2ØØ,32,42,1
        268
CB 430  DATA2ØØ,2Ø1,25,24Ø,67,2
        Ø1,39,2Ø8,242,76,8Ø,195
        ,1774
PB 440  DATA173,251,2Ø1,2Ø8,26,
        162,96,32,82,2ØØ,173,25
        4,1858
EM 450  DATA2Ø1,24Ø,8,2Ø5,97,2Ø
        1,24Ø,3,32,21Ø,255,32,1
        724
RP 460  DATA93,2ØØ,162,117,76,3
        3,197,162,112,32,71,2ØØ
        ,1455
BM 470  DATA162,13Ø,32,71,2ØØ,3
        2,162,2ØØ,32,42,2ØØ,2Ø1
        ,1464
JE 4B0  DATA25,24Ø,2Ø2,2Ø1,39,2
        Ø8,242,169,Ø,133,198,96
        ,1753
GX 490  DATA141,235,2Ø1,169,1,1
        62,B,141,24Ø,2Ø1,142,24
        1,1882
DG 500  DATA2Ø1,32,59,2ØØ,133,1
        98,168,173,24Ø,2Ø1,174,
        241,2Ø2Ø
RP 510  DATA2Ø1,133,253,134,254
        ,32,216,199,177,253,2Ø8
        ,14,2Ø74
EX 520  DATA173,243,2Ø1,2Ø8,155
        ,238,243,2Ø1,141,252,2Ø
        1,76,2332
EC 53Ø DATA6Ø,197,32,216,199,1
        77,253,17Ø,32,216,199,1
        77,1928
JA 54Ø DATA253,142,249,2Ø1,141
        ,25Ø,2Ø1,32,2Ø5,189,169
        ,32,2Ø64
KD 55Ø DATA32,21Ø,255,32,216,1
        99,169,2Ø1,133,252,169,
        96,1964
EP 56Ø DATA174,243,2Ø1,2Ø8,2,1
        69,2Ø8,133,251,162,Ø,14
        2,1893
DK 57Ø DATA228,2Ø1,142,231,2Ø1
        ,142,242,2Ø1,173,232,2Ø
        1,24Ø,2434
KM 58Ø DATA6,142,232,2Ø1,142,2
        52,2Ø1,161,253,24Ø,28,3
        2,189Ø
BJ 59Ø DATA223,199,133,2,32,52
        ,199,165,2,162,Ø,193,13
        62
SR 6ØØ DATA251,2Ø8,99,23Ø,251,
        161,251,24Ø,1Ø2,32,216,
        199,224Ø
RE 61Ø DATA76,175,197,142,216,
        2Ø1,142,244,2Ø1,142,246
        ,2Ø1,2183
CM 62Ø DATA142,252,2Ø1,32,62,2
        ØØ,173,245,2Ø1,2Ø8,16,3
        2,1764
RQ 63Ø DATA216,199,165,253,166
        ,254,141,24Ø,2Ø1,142,24
        1,2Ø1,2419
MG 64Ø DATA76,73,197,142,245,2
        Ø1,173,249,2Ø1,172,25Ø,
        2Ø1,218Ø
FG 65Ø DATA2Ø5,224,2Ø1,2Ø8,5,2
        Ø4,225,2Ø1,24Ø,221,141,
        224,2299
XJ 66Ø DATA2Ø1,14Ø,225,2Ø1,32,
        71,2ØØ,169,19,141,119,2
        ,152Ø
FA 67Ø DATA169,13,141,12Ø,2,14
        1,121,2,141,122,2,169,1
        143
DG 68Ø DATA4,133,198,76,49,168
        ,32,162,2ØØ,32,216,199,
        1469
DD 69Ø DATA76,142,197,142,227,
        2Ø1,142,228,2Ø1,142,233
        ,2Ø1,2132
SC 7ØØ DATA2Ø2,134,2,142,23Ø,2
        Ø1,173,246,2Ø1,2Ø8,227,
        173,2139
HF 71Ø DATA252,2Ø1,32,141,2ØØ,
        164,2,2ØØ,238,23Ø,2Ø1,1
        77,2Ø38
EX 72Ø DATA253,2Ø1,32,24Ø,246,
        132,2,238,227,2Ø1,174,2
        39,2185
RF 73Ø DATA2Ø1,2Ø8,53,2Ø1,4Ø,2
        4Ø,195,32,223,199,173,2
        28,1993
XC 74Ø DATA2Ø1,2Ø8,7,173,252,2
        Ø1,2Ø8,48,24Ø,15,164,2,
        1719
MQ 75Ø DATA2ØØ,177,253,2Ø1,32,
        24Ø,249,132,2,2Ø1,4Ø,24
        Ø,1967
FR 76Ø DATA165,32,24,2ØØ,2Ø5,2
        32,2Ø1,2Ø8,84,173,232,2
        Ø1,1957
JE 77Ø DATA2Ø8,82,2Ø6,23Ø,2Ø1,
        76,224,198,2Ø1,4Ø,24Ø,3
        2,1938
QA 78Ø DATA32,223,199,173,252,
        2Ø1,24Ø,61,173,227,2Ø1,
        2Ø1,2183
DG 79Ø DATA2,144,162,173,97,2Ø
```

```
        1,174,243,2Ø1,2Ø8,3,173
        ,1781
SJ 8ØØ DATA2Ø9,2Ø1,2Ø1,Ø,2Ø8,1
        47,24Ø,37,32,24,2ØØ,2Ø5
        ,17Ø4
RS 81Ø DATA232,2Ø1,2Ø8,29,164,
        2,2ØØ,177,253,24Ø,22,2Ø
        1,1929
8D 82Ø DATA44,2Ø8,3,238,233,2Ø
        1,2Ø1,41,2Ø8,24Ø,2Ø6,23
        Ø,2Ø53
HM 83Ø DATA2Ø1,173,233,2Ø1,2Ø5
        ,234,2Ø1,24Ø,3,76,38,19
        8,2ØØ3
PS 84Ø DATA173,242,2Ø1,32,141,
        2ØØ,172,243,2Ø1,2Ø8,32,
        173,2Ø18
EG 85Ø DATA216,2Ø1,2Ø8,237,169
        ,29,32,53,2ØØ,162,192,3
        2,1731
JS 86Ø DATA82,2ØØ,32,93,2ØØ,16
        2,1Ø2,142,243,2Ø1,32,71
        ,156Ø
HG 87Ø DATA2ØØ,141,252,2Ø1,76,
        234,196,14Ø,245,2Ø1,14Ø
        ,251,2277
HF 88Ø DATA2Ø1,174,236,2Ø1,2Ø2
        ,169,2Ø,32,1Ø3,2ØØ,172,
        23Ø,194Ø
PP 89Ø DATA2Ø1,24Ø,1Ø,48,6,32,
        216,199,136,2Ø8,25Ø,16Ø
        ,17Ø6
KA 9ØØ DATAØ,185,192,2Ø1,24Ø,1
        75,32,12Ø,199,2ØØ,2Ø8,2
        45,1997
FE 91Ø DATA2Ø1,128,144,57,166,
        212,2Ø8,53,2Ø1,131,2Ø8,
        2,1711
RR 92Ø DATA24Ø,4,2Ø1,143,2Ø8,3
        ,141,246,2Ø1,56,233,127
        ,18Ø3
RM 93Ø DATA17Ø,16Ø,255,2Ø2,24Ø
        ,8,2ØØ,185,158,16Ø,16,2
        5Ø,2ØØ4
HC 94Ø DATA 48,245,2ØØ,185,158,
        16Ø,48,14,238,238,2Ø1,3
        2,1767
JE 95Ø DATA113,199,169,Ø,141,2
        38,2Ø1,76,9Ø,199,56,233
        ,1715
EQ 96Ø DATA128,2Ø1,32,24Ø,3,23
        8,242,2Ø1,174,244,2Ø1,4
        8,1952
CC 97Ø DATA11,2Ø8,3Ø,166,211,2
        24,79,144,3,238,244,2Ø1
        ,1759
FH 98Ø DATA174,243,2Ø1,2Ø8,13,
        2Ø1,34,2Ø8,8,173,216,2Ø
        1,188Ø
PK 99Ø DATA73,1,141,216,2Ø1,96
        ,76,21Ø,255,173,238,2Ø1
        ,1881
GH 1ØØØ DATA24Ø,2,1Ø4,1Ø4,1Ø4,
         1Ø4,162,148,142,244,2Ø
         1,32,1587
QS 1Ø1Ø DATA71,2ØØ,174,249,2Ø1
         ,173,25Ø,2Ø1,32,2Ø5,18
         9,169,2114
AE 1Ø2Ø DATA32,32,21Ø,255,173,
         24Ø,2Ø1,174,241,2Ø1,13
         3,253,2145
RF 1Ø3Ø DATA134,254,169,4,133,
         251,164,251,177,253,24
         Ø,14,2Ø44
MR 1Ø4Ø DATA32,52,199,23Ø,251,
         76,2Ø2,199,23Ø,253,2Ø8
         ,2,1934
JG 1Ø5Ø DATA23Ø,254,96,164,212
         ,24Ø,5,16Ø,Ø,76,17,2ØØ
         ,1654
PD 1Ø6Ø DATA2Ø1,32,24Ø,242,2Ø1
```

```
         ,36,24Ø,4,2Ø1,37,2Ø8,9
         ,1651
SK 1Ø7Ø DATA141,228,2Ø1,141,23
         2,2Ø1,76,13,2ØØ,2Ø1,48
         ,144,1826
SC 1Ø8Ø DATA16,2Ø1,58,144,8,2Ø
         1,65,144,8,2Ø1,91,176,
         1313
KJ 1Ø9Ø DATA4,238,252,2Ø1,96,1
         4Ø,232,2Ø1,14Ø,252,2Ø1
         ,96,2Ø53
RB 11ØØ DATA173,255,2Ø1,174,24
         3,2Ø1,24Ø,3,173,254,2Ø
         1,96,2214
HA 111Ø DATA169,4Ø,141,221,2Ø1
         ,96,165,2Ø3,2Ø5,235,2Ø
         1,24Ø,2117
FC 112Ø DATA249,141,235,2Ø1,96
         ,32,21Ø,255,76,21Ø,255
         ,32,1992
SH 113Ø DATA68,229,169,Ø,133,1
         99,133,212,133,216,96,
         189,1777
JP 114Ø DATA177,2ØØ,24Ø,25Ø,32
         ,21Ø,255,232,2Ø8,245,1
         89,Ø,2238
QR 115Ø DATA2Ø1,24Ø,239,32,21Ø
         ,255,232,2Ø8,245,173,2
         39,2Ø1,2475
GX 116Ø DATA24Ø,228,174,234,2Ø
         1,169,42,32,21Ø,255,2Ø
         2,16,2ØØ3
KE 117Ø DATA25Ø,96,173,33,2Ø8,
         76,119,2ØØ,173,134,2,1
         33,1597
SD 118Ø DATA2,162,96,16Ø,5,189
         ,177,2ØØ,153,18,4,165,
         1331
MK 119Ø DATA2,153,18,216,232,1
         36,16,241,96,172,243,2
         Ø1,1726
PA 12ØØ DATA2Ø8,6,2Ø5,237,2Ø1,
         76,155,2ØØ,2Ø5,236,2Ø1
         ,24Ø,217Ø
CJ 121Ø DATA239,1Ø4,1Ø4,76,38,
         198,32,225,255,2Ø8,229
         ,1Ø4,1812
EM 122Ø DATA1Ø4,169,Ø,141,138,
         2,76,68,229,13,83,89,1
         112
HE 123Ø DATA83,53,48,53,48,53,
         Ø,13,13,79,76,68,587
EF 124Ø DATA32,86,65,82,73,65,
         66,76,69,63,32,Ø,7Ø9
AQ 125Ø DATA13,13,78,69,87,32,
         86,65,82,73,65,66,729
AF 126Ø DATA76,69,63,32,Ø,13,1
         3,18,84,89,8Ø,69,6Ø6
RH 127Ø DATA32,77,73,83,77,65,
         84,67,72,46,46,46,768
ME 128Ø DATAØ,13,13,18,65,82,6
         9,32,89,79,85,32,577
KE 129Ø DATA83,85,82,69,63,32,
         4Ø,89,47,78,41,Ø,7Ø9
QP 13ØØ DATA32,32,18,76,73,77,
         73,84,Ø,191,153,129,93
         8
FG 131Ø DATA146,146,129,32,32,
         18,69,88,73,83,84,83,9
         83
FJ 132Ø DATAØ,68,79,78,69,Ø,32
         ,32,18,78,79,84,617
EK 133Ø DATA32,7Ø,79,85,78,68,
         Ø,13,13,67,79,78,662
PA 134Ø DATA84,73,78,85,69,63,
         32,4Ø,89,47,78,41,779
HF 135Ø DATAØ,13,18,69,88,67,6
         9,83,83,73,86,69,718
QK 136Ø DATA32,76,69,78,71,84,
         72,13,13,Ø,5Ø8
```

©

Dr. Sound For The 64

Don Malone

Music enthusiasts will have a field day with this Commodore 64 program, which allows you to experiment with a great variety of different sound parameters while the music plays. A disk drive is required.

"Dr. Sound" is an algorithmic note sequencer which plays notes according to parameters which you choose in realtime. Using the 64's built-in SID (Sound Interface Device) chip, it simulates a single-voice electronic synthesizer with dynamic timbre (tone color) control. If you're familiar with conventional electronic synthesizers, you'll probably recognize the screen display as a flowchart of the synthesizer's current *patch* or configuration. By changing different elements of the patch, you can alter the character of the music dramatically. After you create a patch you like, you can save it to disk for later reloading and use within the program. If you're new to computer-generated music, you'll enjoy experimenting and you can also learn a good deal from this program. Experts will appreciate all the features available in Dr. Sound.

Type in the program as listed and save a copy before you try to run it. Dr. Sound always begins with a short pause while it initializes. Then you will see the main display screen. The top portion of the screen contains a flowchart of the synthesizer's current patch. At the bottom are several prompts indicating parameters you can change by pressing various function keys. The bottom screen line is reserved for your input.

Music In The Background

When the display screen appears, you'll notice that background music begins playing immediately. The music will continue to play at all times while the program runs, except during disk operations.

Using Dr. Sound involves changing various program parameters to alter the character of the music. As a rule, whenever you change the synthesizer's patch, the screen display changes color to indicate which part of the synthesizer you are affecting. The different program options are selected by pressing one of the eight special function keys, f1–f8. Once an option is selected, the bottom screen line changes color and displays the keys you may press to select a choice within that option. In some cases, pressing the indicated key increases the value associated with that parameter; for these options, pressing the SHIFT key along with the indicated key decreases the same value.

Waveform And ADSR

One of the most fundamental changes involves waveforms. To choose a different waveform, press the f1 key. The bottom screen line then indicates your choices. To change the waveform, press the W key. There are four wave shapes available. The triangle is the sweetest of these, containing only odd-numbered overtones decreasing in loudness exponentially. The sawtooth is the brightest, containing all of the harmonic overtones. The pulse wave depends on its width (duty cycle) for its harmonic content. The closer to 99 percent or 1 percent, the more nasal (oboe-like) the pulse wave sounds. The closer to a 50 percent duty cycle, the more hollow (clarinet-like) it will be. Press P to change the pulse width. The noise waveshape is the most unpitched.

Ring modulation is a special SID effect, which you can toggle on and off by pressing the M key. When an *M* appears in the flowchart between the sound source and the modulator, you can see that modulation is on. Ring modulation is possibly the most sophisticated timbre control on the SID chip, making nonharmonic, bell-like overtones. The timbre of the sound depends on the frequency relationship between the sound source and the modulator. (Because of the way the SID chip circuitry is designed, only triangle waveshapes are available for this option.)

Pressing H toggles the harmony option on and off, which forces the sound source to be harmonic—that is, synchronous at an exact integer multiple with the modulator. When the harmony option is selected, an *H* appears in the display between the modulator and the sound source. This can be used to shift the A440 tuning of the sound source or to insure harmonic (more pitched) modulation. *Modulation*—like most of the other terms in this article—can be best understood by listening to the effect it has on different sounds.

The A, D, S, and R keys control attack, decay, sustain, and release, respectively. Attack is the amount of time it takes to begin the note. Decay is the amount of time it takes to drop to the sustain level, which is indicated as a percentage of the loudest sound possible. Release is the amount of time it takes to return to silence.

Special Effects

The f3 key allows you to change the low-pass filter parameters. Q changes the electronic resonance, which at 100 percent almost whistles, indicating sonically the changes in the cutoff frequency. F changes the percentage of the envelope generator (ADSR) used to control the cutoff frequency, and therefore the timbre, during each note. The lower the percentage, the more muffled the sound will be.

The f5 key selects the modulator section. W and P work just like the sound source section. T toggles on and off a trigger that allows the modulator to be heard while also modulating the sound source. I toggles parallel/oblique modes of the interval relationship between the modulator and the sound source. In the parallel mode the frequency follows the sound source at an interval indicated as a percentage of the sound source frequency. M and L change this relationship in 10 percent and 1 percent increments, respectively. Note that there is a delay of about six seconds to calculate these increments. In the oblique mode the frequency of the modulator is always the same. That frequency is tunable from 1 to 3995 Hz (cycles per second). The F, Q, C, and Y keys change the frequency in 1000 Hz, 100 Hz, 10 Hz, and 1 Hz increments, respectively. The ADSR articulation control for the modulator is apparent only when the modulator trigger is on.

The f7 key selects the control section. The W, P, F, Q, C, and Y keys work the same way here as they do in the modulator section. However, in this case the waveshape and the relationship of the frequency to the duration of the current note determine the next note. The triangle and sawtooth waveshapes will produce easily recognizable patterns. The pulse produces a more austere pattern, and the noise waveform produces a random pattern.

G and A change the gate length. During the gate, the attack, decay, and sustain portions of the envelope generators are active. The gate time does not necessarily need to be longer than the attack time plus the decay time, but if it isn't, strange effects, including complete silence, may occur. R and E change the release time. During the release time, the release portion of the envelope generators are active. After the gate and release time, it takes about 223 microseconds to look at the keyboard. This delay becomes much longer if a key has been pressed. It takes another 104–195 microseconds to calculate the next note. However, if the release time of the sound source envelope generator is long enough, these delays will not be apparent.

Pitch Sets

The f2 key allows a choice of one of the 16 pitch sets. The patterns generated by Dr. Sound will be restricted to one of these sets at a time. They are defined as shown here:

0	**Major scale**
1	**Tonic**
2	**Supertonic**
3	**Mediant**
4	**Subdominant**
5	**Dominant**
6	**Submediant**
7	**Diminished**
8	**Subtonic**
9	**Augmented**
A	**Chromatic**
B	**Whole tone**
C	**East**
D	**Harmonic minor**
E	**Pure minor**
F	**Phrygian**

The f4 key allows control over the pitch range. The octaves are labeled from 0 to 7, with octave 0 being the lowest. The octave of each note is chosen from a set of eight possibilities, all of which are displayed on the screen. Pressing a number from 0 to 7 changes the next octave number in the set.

The f6 key allows control over the rhythm. This is also a set of eight, controlled like the octaves. The release time is multiplied by a factor from 1 to 8.

The f8 key permits you to save all of the current Dr. Sound settings with a filename of your choice, or to load a file of previously saved settings.

Dr. Sound For The 64

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
CB 10 POKE53280,0:POKE53281,0:
      PRINTCHR$(142)CHR$(8):PR
      INT"[8]{CLR}":POKE214,10
      :PRINT
DM 20 PRINTTAB(16)"DR. SOUND":
      PRINTTAB(13)"{DOWN}WILL
      {SPACE}8E RUNNING"
RK 30 PRINTTAB(14)"{DOWN}IN 24
       SECONDS":C$=CHR$(13)
FD 40 DIMPI$(11),PT$(15),OC$(7
      ),RH$(7),PI(12),PM(12),A
      $(15),R$(16),S$(15),P$(1
      5)
GS 50 FORC=0TO7:POKE49920+C,4*
      16:NEXT:FORC=0TO7:POKE49
      936+C,1:NEXT:SI=54272
FJ 60 FORC=SITOSI+24:POKEC,0:N
      EXT:POKESI+24,9*16+15:PO
      KE53236,31:POKE53239,128
CR 70 M$(0)="B":M$(1)="M":H$(0
      )="8":H$(1)="H":T$(0)="*
      OFF*":T$(1)="*****"
CG 80 WS$(0)=" TRIANGLE":WS$(1
      )=" SAWTOOTH":WS$(2)="
      {4 SPACES}PULSE":WS$(3)=
      "{2 SPACES}NOISE
      {2 SPACES}"
SR 90 F$(0)="100% ":F$(1)="50%
      {2 SPACES}":F$(2)="25%
      {2 SPACES}":F$(3)="12.5%
      "
MB 100 FORC=50176TO50399:READD
       :POKEC,D:NEXT:FORC=0TO1
       2:READD:PI(C)=D:NEXT
AR 110 FORI=0TO15:READPT$(I):F
       ORC=0TO15:READD:POKE496
       64+I*16+C,D:NEXT:NEXT
MF 120 FORC=0TO15:READA$(C):NE
       XT:FORC=0TO16:READR$(C)
       :NEXT:FORC=0TO7:E(C)=2↑
       C:NEXT
QQ 130 FORC=0TO15:S$(C)=STR$(I
       NT(C*6.66666667)):S$(C)
       =S$(C)+"%{2 SPACES}":NE
       XT
KJ 140 FORC=0TO15:P$(C)=STR$(I
       NT(C*256/40.95)):P$(C)=
       P$(C)+"?% ":NEXT
KR 150 ML=.99:GOSU8550:GOSUB62
       0:CW=1:H=0:M=0:PC=0:AC=
          Ø:DC=Ø:SC=15:RC=4:GOSUB
          71Ø
XK 16Ø  RS=4:FC=1:AF=7:DF=3:SF=
          13:RF=5:GOSUB78Ø:PW=Ø:P
          C=Ø:AP=Ø:DP=Ø:SP=8:RP=1
          5
SD 17Ø  TP=1:FM=1:FP=Ø:GOSUB83Ø
          :WW=1:FQ=112:GG=2Ø:RR=2
          2:GOSUB94Ø:PT=13
FP 18Ø  IN$="TRUMPET":POKE53232
          ,PT*16:GOSUB1Ø5Ø:GOSUB1
          Ø8Ø:GOSUB112Ø:GOSUB115Ø
PB 19Ø  SYS5Ø176:GETK$:IFK$<>""
          THEN117Ø
FB 2ØØ  GOTO19Ø
QF 21Ø  DATA 162,Ø,173,27,212,4
          1,7,17Ø,189,16,195,17Ø,
          173,244,2Ø7,2Ø2
JP 22Ø  DATA 48,6,1Ø9,244,2Ø7,7
          6,15,196,141,243,2Ø7,16
          2,Ø,173,27,212
DP 23Ø  DATA 41,15,141,239,2Ø7,
          173,24Ø,2Ø7,1Ø9,239,2Ø7
          ,17Ø,189,Ø,194,17Ø
QC 24Ø  DATA 173,27,212,41,7,16
          8,185,Ø,195,141,238,2Ø7
          ,138,1Ø9,238,2Ø7
XH 25Ø  DATA 17Ø,189,Ø,192,141,
          7,212,189,128,192,141,8
          ,212,173,248,2Ø7
GA 26Ø  DATA 2Ø8,12,189,Ø,193,1
          41,Ø,212,189,128,193,14
          1,1,212,173,252
BG 27Ø  DATA 2Ø7,1Ø5,1,141,18,2
          12,173,251,2Ø7,1Ø5,1,14
          1,11,212,173,25Ø
SB 28Ø  DATA 2Ø7,1Ø9,249,2Ø7,14
          1,4,212,173,247,2Ø7,141
          ,255,2Ø7,173,246,2Ø7
KC 29Ø  DATA 141,237,2Ø7,172,24
          5,2Ø7,173,28,212,174,25
          3,2Ø7,24Ø,4,74,2Ø2
HB 3ØØ  DATA 2Ø8,252,141,22,212
          ,136,2Ø8,238,2Ø6,237,2Ø
          7,2Ø8,23Ø,2Ø6,255,2Ø7
MS 31Ø  DATA 2Ø8,219,173,252,2Ø
          7,141,18,212,173,251,2Ø
          7,141,11,212,173,25Ø
XQ 32Ø  DATA 2Ø7,141,4,212,173,
          243,2Ø7,141,254,2Ø7,173
          ,242,2Ø7,141,237,2Ø7
PH 33Ø  DATA 172,241,2Ø7,173,28
          ,212,174,253,2Ø7,24Ø,4,
          74,2Ø2,2Ø8,252,141
KF 34Ø  DATA 22,212,136,2Ø8,238
          ,2Ø6,237,2Ø7,2Ø8,23Ø,2Ø
          6,254,2Ø7,2Ø8,219,96
XG 35Ø  DATA268,284,3Ø1,318,337
          ,358,379,4Ø1,425,451,47
          7,5Ø6,536
BQ 36Ø  DATA"MAJOR{10 SPACES}",
          5,7,9,1Ø,12,5,4,2,Ø,Ø,1
          2,9,5,9,5,Ø
XS 37Ø  DATA"TONIC{10 SPACES}",
          5,9,12,Ø,5,9,12,Ø,5,9,1
          2,Ø,5,9,12,Ø
GR 38Ø  DATA"SUPERTONIC 7TH ",7
          ,1Ø,2,5,7,1Ø,2,5,7,1Ø,2
          ,5,7,1Ø,2,5
HP 39Ø  DATA"MEDIANT{8 SPACES}"
          ,9,12,Ø,3,9,12,Ø,3,9,12
          ,Ø,3,9,12,Ø,3
XP 4ØØ  DATA"SUBDOMINANT
          {4 SPACES}",1Ø,2,5,1Ø,2
          ,5,1Ø,2,5,1Ø,2,5,1Ø,2,5
          ,1Ø
HS 41Ø  DATA"DOMINANT 7TH
          {3 SPACES}",Ø,4,7,1Ø,12
          ,Ø,4,7,1Ø,12,Ø,4,7,1Ø,1
          2,Ø
KC 42Ø  DATA"SUBMEDIANT
          {5 SPACES}",2,5,9,12,Ø,
          2,5,9,2,5,9,2,5,9,12,Ø
MC 43Ø  DATA"DIMINISHED 7TH ",4
          ,7,1Ø,1,4,7,1Ø,1,4,7,1Ø
          ,1,4,7,1Ø,1
DJ 44Ø  DATA"SUBTONIC{7 SPACES}
          ",3,7,1Ø,3,7,1Ø,3,7,1Ø,
          3,7,1Ø,3,7,1Ø,3
CE 45Ø  DATA"AUGMENTED
          {6 SPACES}",Ø,4,8,12,Ø,
          4,8,12,Ø,4,8,12,Ø,4,8,1
          2
MG 46Ø  DATA"CHROMATIC
          {6 SPACES}",Ø,1,2,3,4,5
          ,6,7,8,9,1Ø,11,12,Ø,12,
          6
MJ 47Ø  DATA"WHOLETONE
          {6 SPACES}",Ø,2,4,6,8,1
          Ø,12,1Ø,8,6,4,2,Ø,2,1Ø,
          12
KD 48Ø  DATA"EAST{11 SPACES}",Ø
          ,2,5,7,9,12,Ø,2,5,7,9,1
          2,5,2,5,7
EE 49Ø  DATA"HARMONIC MINOR ",5
          ,7,8,1Ø,12,1Ø,8,7,5,4,1
          ,Ø,1,4,5,8
CA 5ØØ  DATA"PURE MINOR
          {5 SPACES}",5,7,8,1Ø,12
          ,1Ø,8,7,5,3,1,Ø,1,3,5,8
JQ 51Ø  DATA"PHRYGIAN{7 SPACES}
          ",5,6,8,1Ø,12,1Ø,8,6,5,
          3,2,Ø,2,3,5,8
XF 52Ø  DATA.ØØ2S,.ØØ8S,.Ø16S,.
          Ø24S,.Ø38S,.Ø56S,.Ø68S,
          .Ø8S ,.1S{2 SPACES},.25
          S ,.5S{2 SPACES},.8S
GE 53Ø  DATA1S{3 SPACES},3S
          {3 SPACES},5S{3 SPACES}
          ,8S{3 SPACES},.ØØ6S,.Ø2
          4S,.Ø48S,.Ø72S,.114S,.1
          68S,.2Ø4S
PS 54Ø  DATA.24S ,.3S{2 SPACES}
          ,.75S ,1.5S ,2.4S ,3S
          {3 SPACES},9S{3 SPACES}
          ,15S{2 SPACES},"24S
          {2 SPACES}","{7 SPACES}
          "
EK 55Ø  FORC=ØTO12:FORI=ØTO7:PI
          =PI(C)*E(I):HP=INT(PI/2
          56):IFHP>255THENHP=255
EM 56Ø  POKE4928Ø+I*16+C,HP:POK
          E49152+I*16+C,PI-256*HP
          AND255:NEXT:NEXT
GP 57Ø  IFLEN(STR$(ML))>5THENML
          =INT(ML*1ØØ)/1ØØ
CP 58Ø  FORC=ØTO12:PM(C)=PI(C)*
          ML:NEXT
BX 59Ø  FORC=ØTO12:FORI=ØTO7:PI
          =PM(C)*E(I):HP=INT(PI/2
          56):IFHP>255THENHP=255
XP 6ØØ  POKE49536+I*16+C,HP:Z=P
          I-256*HP:IFZ>255THENZ=2
          55
SB 61Ø  POKE494Ø8+I*16+C,Z:NEXT
          :NEXT:RETURN
CE 62Ø  PRINT"{CLR}{2 DOWN}"SPC
          (1Ø)"CCC>AMPC*CCCC{S}"
BP 63Ø  PRINTTAB(4)"↑"SPC(1Ø)"↑
          "SPC(7)"B"
MB 64Ø  PRINTTAB(4)"B"SPC(18)"B
          "
GB 65Ø  PRINTTAB(4)"B"SPC(18)"B
          "
JQ 66Ø  PRINTTAB(4)"B"SPC(18)"
          {Q}**>FILTER**>OUT"
CR 67Ø  PRINTTAB(4)"B"SPC(18)"B
          "SPC(5)"↑"
MQ 68Ø  PRINTTAB(4)"B"SPC(18)"B
          "
GQ 69Ø  PRINTTAB(23)"B"SPC(5)"↑
          ":PRINTTAB(9)"*****>AMP
          "SPC(5)"{X}":PRINTTAB(1
          6)"↑"
RA 7ØØ  RETURN
CE 71Ø  PRINT"{HOME}{DOWN}":IFM
          =1THENCW=Ø
QH 72Ø  PRINTTAB(2)WS$(CW):IFCW
          =2THENPRINT" {UP}"P$(PC
          )
SG 73Ø  PRINTTAB(7)"{DOWN}[F1]"
          SPC(3)"A "A$(AC):PRINTT
          AB(4)M$(M)SPC(9)"D "R$(
          DC)
XP 74Ø  PRINTTAB(14)"S"S$(SC):P
          RINTTAB(4)H$(H)SPC(9)"R
           "R$(RC)
PP 75Ø  WC=2↑(CW+4):IFM=1THENWC
          =2Ø
SA 76Ø  IFH=1THENWC=WC+2
CD 77Ø  POKE53243,WC:POKESI+1Ø,
          PC:POKESI+12,AC*16+DC:P
          OKESI+13,SC*16+RC:RETUR
          N
JE 78Ø  POKE214,4:PRINT:PRINTTA
          B(28)"Q"S$(RS)
SG 79Ø  PRINTTAB(28)"{2 DOWN}"F
          $(FC):PRINTTAB(34)"{UP}
          [F3]"
BJ 8ØØ  PRINTTAB(28)"{DOWN}A "A
          $(AF):PRINTTAB(28)"D "R
          $(DF)
FC 81Ø  PRINTTAB(28)"S"S$(SF):P
          RINTTAB(28)"R "R$(RF)
CB 82Ø  POKESI+23,RS*16+3:POKE5
          3245,FC:POKESI+19,AF*16
          +DF:POKESI+2Ø,SF*16+RF:
          RETURN
AH 83Ø  POKE214,8:PRINT:IFM=1TH
          ENPW=Ø
CS 84Ø  PRINT"{DOWN} "WS$(PW)SP
          C(8)T$(TP):IFPW=2THENPR
          INT"{UP}"P$(PP)
QS 85Ø  IFFP=1THENPRINT"
          {4 SPACES}FQCY
          {4 SPACES}":PRINTTAB(4)
          STR$(FM)+"HZ{3 SPACES}"
EJ 86Ø  IFFP=1THENZ=FM/.Ø6Ø97:P
          H=INT(Z/256):PL=Z-PH*25
          6:POKESI,PL:POKESI+1,PH
JA 87Ø  IFFP=ØTHENPRINT"
          {4 SPACES}PARALLEL":PRI
          NTTAB(4)"ML"STR$(INT(ML
          *1ØØ+.5))+"%{2 SPACES}"
GC 88Ø  IFTP=1THENPRINTTAB(15)"
          {UP}A "A$(AP):PRINTTAB(
          15)"D "R$(DP)
RH 89Ø  IFTP=ØTHENPRINTTAB(15)"
          {UP}"R$(16):PRINTTAB(15
          )R$(16)
PF 9ØØ  IFTP=1THENPRINTTAB(9)"[
          F5]"SPC(2)"S"S$(SP):PRI
          NTTAB(15)"R "R$(RP)
SA 91Ø  IFTP=ØTHENPRINTTAB(9)"[
          F5]"SPC(2)R$(16):PRINTT
          AB(15)R$(16)
XQ 92Ø  WP=2↑(PW+4):POKE53242,W
          P:POKESI+3,PP:POKE53241
          ,TP:POKE5324Ø,FP
FQ 93Ø  POKESI+5,AP*16+DP:POKES
          I+6,SP*16+RP:RETURN
QC 94Ø  POKE214,16:PRINT
EC 95Ø  PRINT"[F7] "WS$(WW)SPC(
          1)"FQCY"STR$(FQ)+"HZ
          {3 SPACES}"
RP 96Ø  IFWW=2THENPRINTTAB(4)"
          {UP}"P$(CP)
RF 97Ø  SS=2↑(WW+4):POKE53244,S
          S:POKESI+17,CP:Z=FQ/.Ø6
```

```
         Ø97:CH=INT(Z/256):CL=Z-
         CH*256
FR 98Ø  POKESI+14,CL:POKESI+15,
         CH:GY=(GGAND127)+1:GL=1
         :IFGG>127THENGL=128
GG 99Ø  GT=(4Ø+(((8+(((4+(((8+(
         (7*FC)+2)+9)*GY)-1)+9)*
         GL)-1)+9)*128)-1))/1Ø2Ø
         ØØØ
FF 1ØØØ GT=INT(GT*1ØØØ)/1ØØØ:P
         RINTTAB(27)"{2 UP}GA"S
         TR$(GT)+"S{2 SPACES}"
CE 1Ø1Ø RY=(RRAND127)+1:RL=1:I
         FRR>127THENRL=128
FG 1Ø2Ø RT=(32+(((8+(((4+(((8+
         ((7*FC)+2)+9)*RY)-1)+9
         )*RL)-1)+9)*31)-1))/1Ø
         2ØØØØ
KP 1Ø3Ø RT=INT(RT*1ØØØ)/1ØØØ:P
         RINTTAB(27)"RE"STR$(RT
         )+"S{2 SPACES}"
EX 1Ø4Ø POKE53238,GL:POKE53237
         ,GY:POKE53234,RL:POKE5
         3233,RY:RETURN
CS 1Ø5Ø POKE214,18:PRINT:PRINT
         "[F2] PITCH SET(Ø-F)";
         :IFPT<1ØTHENPRINTPT;PT
         $(PT)
GD 1Ø6Ø IFPT>9THENPRINT" "CHR$
         (PT+55)" "PT$(PT)
FP 1Ø7Ø POKE53232,PT*16:RETURN
RJ 1Ø8Ø POKE214,19:PRINT
AE 1Ø9Ø FORC=ØTO7:OC$(C)=RIGHT
         $(STR$((PEEK(4992Ø+C))
         /16),1):NEXT
AD 11ØØ PRINT"[F4] OCTAVE (Ø-7
         ) ";:FORC=ØTO7:PRINTOC
         $(C);CHR$(44);:NEXT:PR
         INT"{LEFT} "
CE 111Ø RETURN
SF 112Ø POKE214,2Ø:PRINT:FORC=
         ØTO7:RH$(C)=RIGHT$(STR
         $((PEEK(49936+C))+1),1
         ):NEXT
PG 113Ø PRINT"[F6] RHYTHM (1-8
         ) ";:FORC=ØTO7:PRINTRH
         $(C);CHR$(44);:NEXT:PR
         INT"{LEFT} {HOME}"
KG 114Ø RETURN
XA 115Ø POKE214,21:PRINT:PRINT
         "[F8] DISK ACCESS
         {HOME}"
FX 116Ø POKE214,Ø:PRINT:PRINTT
         AB(27)IN$:RETURN
KX 117Ø K=ASC(K$+CHR$(Ø)):IFK>
         =132ANDK<=14ØTHENGOSUB
         119Ø
GE 118Ø ONJGOSUB123Ø,138Ø,152Ø
         ,18ØØ,2Ø4Ø,2Ø8Ø,211Ø,2
         14Ø:GOTO19Ø
KS 119Ø IFJ=ØTHEN121Ø
RK 12ØØ PRINT"{8}":ONJGOSUB123
         Ø,138Ø,152Ø,18ØØ,2Ø4Ø,
         2Ø8Ø,211Ø,214Ø
XP 121Ø J=K-132:PRINT"{4}":RET
         URN
RR 122Ø RETURN
HJ 123Ø POKE214,22:PRINT:PRINT
         "{RVS} SOUND SOURCE
         {5 SPACES}W P M H A D
         {SPACE}S R{6 SPACES}
         {OFF}"
JQ 124Ø IFK$="W"THENCW=(CW+1)A
         ND3
PP 125Ø IFK$="H"THENH=H+1AND1
EP 126Ø IFK$="M"THENM=M+1AND1
AM 127Ø IFK$="P"THENPC=PC+1AND
         15
JA 128Ø IFK$="P"THENPC=ABS(PC-
         1)
```

```
EH 129Ø IFK$="A"THENAC=AC+1AND
         15
FX 13ØØ IFK$="D"THENDC=DC+1AND
         15
KB 131Ø IFK$="S"THENSC=SC+1AND
         15
EC 132Ø IFK$="R"THENRC=RC+1AND
         15
E8 133Ø IFK$="A"THENAC=ABS(AC-
         1)
QE 134Ø IFK$="D"THENDC=ABS(DC-
         1)
CF 135Ø IFK$="S"THENSC=ABS(SC-
         1)
FH 136Ø IFK$="R"THENRC=ABS(RC-
         1)
SJ 137Ø K$="":GOTO71Ø
SB 138Ø POKE214,22:PRINT:PRINT
         "{RVS} FILTER
         {4 SPACES}Q F A D S R
         {17 SPACES}{OFF}"
BG 139Ø IFK$="Q"THENRS=RS+1AND
         15
MP 14ØØ IFK$="Q"THENRS=ABS(RS-
         1)
8P 141Ø IFK$="F"THENFC=FC+1AND
         3
AP 142Ø IFK$="F"THENFC=ABS(FC-
         1)
CG 143Ø IFK$="A"THENAF=AF+1AND
         15
FQ 144Ø IFK$="D"THENDF=DF+1AND
         15
F8 145Ø IFK$="S"THENSF=SF+1AND
         15
GD 146Ø IFK$="R"THENRF=RF+1AND
         15
CX 147Ø IFK$="A"THENAF=ABS(AF-
         1)
MC 148Ø IFK$="D"THENDF=ABS(DF-
         1)
ED 149Ø IFK$="S"THENSF=ABS(SF-
         1)
KH 15ØØ IFK$="R"THENRF=A8S(RF-
         1)
RP 151Ø K$="":GOTO78Ø
EJ 152Ø POKE214,22:PRINT:PRINT
         "{RVS} MODULATOR W P T
          I M/L F/Q/C/Y A D S R
          {OFF}"
JQ 153Ø IFK$="W"THENPW=(PW+1)A
         ND3
FM 154Ø IFK$="P"THENPP=PP+1AND
         15
BX 155Ø IFK$="P"THENPP=ABS(PP-
         1)
MP 156Ø IFK$="T"THENTP=TP+1AND
         1
SA 157Ø IFK$="I"THENFP=FP+1AND
         1
AH 158Ø IFK$="F"THENFM=FM+1ØØØ
RK 159Ø IFK$="Q"THENFM=FM+1ØØ
PD 16ØØ IFK$="C"THENFM=FM+1Ø
SP 161Ø IFK$="Y"THENFM=FM+1
HR 162Ø IFFM>3995THENFM=3995
CF 163Ø IFK$="F"THENFM=ABS(FM-
         1ØØØ)
8K 164Ø IFK$="Q"THENFM=ABS(FM-
         1ØØ)
8P 165Ø IFK$="C"THENFM=ABS(FM-
         1Ø)
HD 166Ø IFK$="Y"THENFM=ABS(FM-
         1)
RK 167Ø IFK$="M"THENML=ML+.1:G
         OSUB57Ø
RJ 168Ø IFK$="L"THENML=ML+.Ø1:
         GOSU857Ø
8D 169Ø IFK$="M"THENML=ABS(ML-
         .1):GOSUB57Ø
DA 17ØØ IFK$="L"THENML=ABS(ML-
         .Ø1):GOSUB57Ø
```

```
MQ 171Ø IFK$="A"THENAP=AP+1AND
         15
MC 172Ø IFK$="D"THENDP=DP+1AND
         15
AH 173Ø IFK$="S"THENSP=SP+1AND
         15
GK 174Ø IFK$="R"THENRP=RP+1AND
         15
DD 175Ø IFK$="A"THENAP=ABS(AP-
         1)
KG 176Ø IFK$="D"THENDP=ABS(DP-
         1)
BF 177Ø IFK$="S"THENSP=ABS(SP-
         1)
KJ 178Ø IFK$="R"THENRP=ABS(RP-
         1)
RM 179Ø K$="":GOTO83Ø
RM 18ØØ POKE214,22:PRINT:PRINT
         "{RVS} CONTROL
         {4 SPACES}W P F/Q/C/Y
         {2 SPACES}G/A
         {2 SPACES}R/E
         {6 SPACES}{OFF}"
JM 181Ø IFK$="W"THENWW=(WW+1)A
         ND3
SG 182Ø IFK$="P"THENCP=CP+1AND
         15
XQ 183Ø IFK$="P"THENCP=ABS(CP-
         1)
PK 184Ø IFK$="F"THENFQ=FQ+1ØØØ
RD 185Ø IFK$="Q"THENFQ=FQ+1ØØ
PH 186Ø IFK$="C"THENFQ=FQ+1Ø
GS 187Ø IFK$="Y"THENFQ=FQ+1
XF 188Ø IFFQ>3995THENFQ=3995
RJ 189Ø IFK$="F"THENFQ=ABS(FQ-
         1ØØØ)
KS 19ØØ IFK$="Q"THENFQ=ABS(FQ-
         1ØØ)
KB 191Ø IFK$="C"THENFQ=ABS(FQ-
         1Ø)
DH 192Ø IFK$="Y"THENFQ=ABS(FQ-
         1)
PJ 193Ø IFK$="G"THENGG=GG+25
MQ 194Ø IFK$="A"THENGG=GG+1
QM 195Ø IFGG>255THENGG=255
XJ 196Ø IFK$="G"THENGG=ABS(GG-
         25)
MR 197Ø IFK$="A"THENGG=ABS(GG-
         1)
BK 198Ø IFK$="R"THENRR=RR+25
XH 199Ø IFK$="E"THENRR=RR+1
KG 2ØØØ IFRR>255THENRR=255
PF 2Ø1Ø IFK$="R"THENRR=ABS(RR-
         25)
AS 2Ø2Ø IFK$="E"THENRR=ABS(RR-
         1)
QF 2Ø3Ø K$="":GOTO94Ø
QD 2Ø4Ø POKE214,22:PRINT:PRINT
         "{RVS} PITCH SET
         {2 SPACES}Ø 1 2 3 ...
         {SPACE}9 A B C D E F
         {2 SPACES}{OFF}"
HE 2Ø5Ø IFK<58ANDK>47THENPT=K-
         48
PM 2Ø6Ø IFK<71ANDK>64THENPT=K-
         55
GJ 2Ø7Ø K$="":GOTO1Ø5Ø
KR 2Ø8Ø POKE214,22:PRINT:PRINT
         "{RVS} OCTAVES
         {4 SPACES}Ø,1,2,3,4,5,
         6,7{12 SPACES}{OFF}"
DJ 2Ø9Ø IFK<56ANDK>47THENK=K-4
         8:CT=CT+1AND7:POKE4992
         Ø+CT,K*16
JQ 21ØØ K$="":GOTO1Ø8Ø
AC 211Ø POKE214,22:PRINT:PRINT
         "{RVS} RHYTHMS
         {6 SPACES}1,2,3,4,5,6,
         7,8{1Ø SPACES}{OFF}"
RA 212Ø IFK<57ANDK>48THENK=K-4
         9:YT=YT+1AND7:POKE4993
```

```
              6+YT,K
KK 2130 K$="":GOTO1120
AS 2140 POKE214,22:PRINT:PRINT
        "{RVS}{2 SPACES}DISK A
        CCESS{5 SPACES}S L
        {18 SPACES}{OFF}"
SC 2150 IFK$="S"THENGOSUB2180
FR 2160 IFK$="L"THENGOSUB2300:
        GOSUB2440
BD 2170 K$="":GOTO1150
AA 21B0 POKE214,22:PRINT:PRINT
        "{RVS} SAVE FILE NAME
        {21 SPACES}{OFF}"
BS 2190 PRINTTAB(18)"{UP}{RVS}
        ";:INPUTIN$:PRINT"{UP}
        {OFF}":IN$=LEFT$(IN$,1
        2)
AX 2200 OPEN15,8,15:OPEN2,8,2,
        "0:"+IN$+",S,W"
QS 2210 GOSUB2410:IFEN>1THENFO
        RC=0TO5000:NEXT:CLOSE2
        :CLOSE15:RETURN
CD 2220 PRINT#2,CW;C$;H;C$;M;C
        $;PC;C$;AC;C$;DC;C$;SC
        ;C$;RC
EG 2230 PRINT#2,RS;C$;FC;C$;AF
        ;C$;DF;C$;SF;C$;RF
AH 2240 MD=ML:PRINT#2,PW;C$;PP
        ;C$;MD;C$;FM;C$;TP;C$;
        FP;C$;AP;C$;DP;C$;SP;C
        $;RP
SQ 2250 PRINT#2,WW;C$;CP;C$;FQ
        ;C$;GG;C$;RR;C$;PT:GOS
        UB2410
GF 2260 FORC=0TO7:PRINT#2,PEEK
        (49920+C)
GJ 2270 NEXT:FORC=0TO7:PRINT#2
        ,PEEK(49936+C)
ER 22B0 NEXT:GOSUB2410
XQ 2290 CLOSE2:CLOSE15:RETURN
EJ 2300 POKE214,22:PRINT:PRINT
        "{RVS} LOAD FILE NAME
        {21 SPACES}{OFF}"
FA 2310 PRINTTAB(18)"{UP}{RVS}
        ";:INPUTIN$:PRINT"{UP}
        {OFF}":IN$=LEFT$(IN$,1
        2)
JP 2320 OPEN15,8,15:OPEN2,8,2,
        "0:"+IN$+",S,R"
RG 2330 GOSUB2410:IFEN>1THENFO
        RC=0TO5000:NEXT:CLOSE2
        :CLOSE15:RETURN
BF 2340 INPUT#2,CW,H,M,PC,AC,D
        C,SC,RC
GK 2350 INPUT#2,RS,FC,AF,DF,SF
        ,RF
JX 2360 INPUT#2,PW,PP,MD,FM,TP
        ,FP,AP,DP,SP,RP
SE 2370 INPUT#2,WW,CP,FQ,GG,RR
        ,PT:GOSUB2410
RQ 2380 FORC=0TO7:INPUT#2,X:PO
        KE(49920+C),X:NEXT
QP 2390 FORC=0TO7:INPUT#2,X:PO
        KE(49936+C),X:NEXT:GOS
        UB2410
HG 2400 CLOSE2:CLOSE15:RETURN
KG 2410 INPUT#15,EN,EM$,ET,ES
SH 2420 IFEN>1THENPOKE214,22:P
        RINT:PRINTCHR$(1B);EM$
        ;CHR$(32);"{5 SPACES}"
CK 2430 RETURN
RK 2440 IFFP=0THENIFMD<>MLTHEN
        ML=MD:GOSUB570
RM 2450 PRINT"[8]{HOME}":GOSUB
        710:GOSUB7B0:GOSUBB30:
        GOSUB940
EG 2460 GOSUB1050:GOSUB1080:GO
        SUB1120:PRINT"[4]
        {HOME}":GOSUB1150:RETU
        RN                    ©
```

Fast Data For 64

Bob Kododek

This handy Commodore 64 routine offers a speedy alternative to READing large amounts of information from DATA statements and POKEing it into memory. By using this automatic technique, you can cut program initialization delays dramatically. Use it for new programs or convert all your old ones—either way, you'll be delighted at the difference it makes.

Have you ever waited for a BASIC program to READ loads of data from DATA statements and POKE it into memory? This has always been the traditional way to store data for sprite images or custom characters, to set up musical note tables, and for many other purposes. No matter what the goal, there are few experiences more tedious than staring at a PLEASE WAIT message while BASIC executes hundreds (or even thousands) of READ and POKE statements. "Fast Data For 64" can perform such operations in a flash, at the speed of machine language. Yet, it becomes part of your BASIC program and is simply called with a GOSUB. For example, 2000 bytes of data can be read and POKEd into memory in only 6/10 second—about 3000 bytes per second. It takes BASIC over 27 full seconds to do the same job. Best of all, this routine automatically appends itself to any BASIC program and can be used even if you don't know anything about machine language.

A Speedy Alternative

Type in and save the program as it appears in the listing. When you run it, the program installs a machine language routine in memory, then displays several instructions on the screen. Next, load the BASIC program you wish to convert. After the load is finished, enter SYS 49152 and press RETURN. When the word LIGHTNING appears on the screen, a special routine has been added to your program. If you list the program, you will notice that it now contains four extra lines, numbered 63996–63999. (These line numbers are used because the routine must be located at the very end of your program, and BASIC will not allow line numbers higher than 63999.)

Now locate the very last DATA statement in your program and add a comma followed by −1. For instance, say that the last DATA line in the program looks like this:

```
5000 DATA 224,169,255,96
```

You'd change it to:

```
5000 DATA 224,169,255,96,−1
```

The value −1 marks the end of the data. (Because −1 is used as a marker, you cannot use this program for data that contains the value −1 elsewhere. This shouldn't pose any problems when the program is used for its intended purpose, since it's impossible to POKE a negative value into a memory location.)

To call the routine, add a line which sets the variable D equal to the beginning of the memory area where you want to store the data and then executes GOSUB 63997. For example, to move a block of data into screen memory, which normally begins at location 1024, you could use this line:

```
100 D=1024:GOSUB 63997
```

The same procedure is used whether you're writing a new program or enhancing an existing one. If you're updating an existing program, be sure to remove the old lines that previously did the POKEing. (Of course, you must not remove the DATA lines themselves, since the ML routine still needs something to read.) This routine uses the variable names D, D%, and A, so you must not use those variables anywhere in your own program. When you're finished making the changes, save the modified version of the program with a new filename.

If you're interested in how all this works: Line 63997 of the conversion routine changes the variable D into a low-byte/high-byte address and sets up a pointer at 253–254 ($FD–$FE) for the machine language routine to use in storing the data. Line 63998 updates the DATA pointer at 65 ($41) by reading and POKEing the first byte of data from BASIC. It then calculates the location of the machine language routine in BASIC memory and calls it with the resultant SYS number. Line 63999 contains the actual machine language in a REM statement. This technique works fine as long as the code is relocatable and does not contain any zero bytes or control characters. Note that this special line contains more than the usual 80 characters. Do *not* attempt to edit or change this line in any way; the BASIC editor will shorten the line and scramble the machine language it contains.

Fast Data For 64

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
XB 1Ø PRINT"{CLR}{DOWN}PLEASE
      {SPACE]WAIT":FOR I=Ø TO
      {SPACE}386:READ 8Y:POKE
      {SPACE}49152+I,8Y:CK=CK+
      8Y:NEXT
RF 2Ø IF CK <> 38541 THEN PRIN
      T"ERROR IN DATA STATEMEN
      T!":END
GG 3Ø DATA 162,Ø,189,1Ø1,193,2
      4Ø,6,32
XC 4Ø DATA 21Ø,255,232,2Ø8,245
      ,169,77,133
GS 5Ø DATA 17Ø,169,192,133,171
      ,32,51,165
FQ 6Ø DATA 16Ø,Ø,177,17Ø,2Ø1,3
      ,24Ø,18
EM 7Ø DATA 145,34,23Ø,34,2Ø8,2
      ,23Ø,35
SD 8Ø DATA 23Ø,17Ø,2Ø8,2,23Ø,1
      71,16Ø,Ø
MG 9Ø DATA 24Ø,232,32,51,165,1
      65,34,24
QF 1ØØ DATA 1Ø5,2,144,2,23Ø,35
       ,133,45
FJ 11Ø DATA 133,47,133,49,165,
       35,133,46
GA 12Ø DATA 133,48,133,5Ø,96,5
       8,8,252
MR 13Ø DATA 249,128,58,143,32,
       82,38,8Ø
SG 14Ø DATA 32,82,79,85,84,73,
       78,69
BB 15Ø DATA 46,7Ø,73,82,83,84,
       32,83
JP 16Ø DATA 69,84,32,68,61,84,
       79,32
QR 17Ø DATA 68,69,83,84,32,84,
       72,69
FM 18Ø DATA 78,32,71,79,83,85,
       66,32
DC 19Ø DATA 54,51,57,57,55,Ø,1
       Ø3,8
DR 2ØØ DATA 253,249,68,37,178,
       68,173,5Ø
KA 21Ø DATA 53,54,58,151,5Ø,53
       ,52,44
QX 22Ø DATA 68,37,58,151,5Ø,53
       ,51,44
CB 23Ø DATA 68,171,68,37,172,5
       Ø,53,54
PM 24Ø DATA 58,135,32,65,58,15
       1,32,68
P8 25Ø DATA 44,65,Ø,156,8,254,
       249,158
RD 26Ø DATA 32,194,4Ø,52,54,41
       ,172,5Ø
JX 27Ø DATA 53,54,17Ø,194,4Ø,5
       2,53,41
RJ 28Ø DATA 171,32,49,49,56,32
       ,58,142
KK 29Ø DATA 58,143,32,7Ø,73,78
       ,68,32
CX 3ØØ DATA 49,83,84,32,66,89,
       84,69
FK 31Ø DATA 32,79,7Ø,32,77,47,
       76,Ø
KG 32Ø DATA 22,9,255,249,143,3
       4,23Ø,253
XR 33Ø DATA 2Ø8,2,23Ø,254,16Ø,
       255,2ØØ,132
GP 34Ø DATA 98,132,99,132,1ØØ,
       23Ø,65,2Ø8
MS 35Ø DATA Ø2,23Ø,66,177,65,2
       Ø8,Ø14,165
CF 36Ø DATA 65,24,1Ø5,5,133,65
       ,144,44
MM 37Ø DATA 23Ø,66,2Ø8,4Ø,234,
       2Ø1,44,24Ø
EK 38Ø DATA 35,2Ø1,32,24Ø,224,
       2Ø1,45,2Ø8
FK 39Ø DATA 12,165,65,24,1Ø5,2
       ,133,65
RX 4ØØ DATA 144,2,23Ø,66,96,56
       ,233,48
QE 41Ø DATA 166,99,134,98,166,
       1ØØ,134,99
FP 42Ø DATA 133,1ØØ,176,193,16
       2,1ØØ,165,98
FC 43Ø DATA 24Ø,9,2Ø1,1,24Ø,2,
       162,2ØØ
8F 44Ø DATA 138,133,98,165,99,
       24Ø,8,162
XX 45Ø DATA 9,24,1Ø1,99,2Ø2,2Ø
       8,25Ø,24
QR 46Ø DATA 1Ø1,98,24,1Ø1,1ØØ,
       145,253,144
JE 47Ø DATA 141,Ø,Ø,Ø,3,76,73,
       71
AF 48Ø DATA 72,84,78,73,78,71,
       33,Ø13
FJ 49Ø DATA Ø,4Ø,67,41,49,57,5
       6,54
DC 5ØØ DATA 66,79,66,75,79,68,
       65,68,69,75,Ø
```

©

Enhancements For Atari *SpeedCalc*

Fred Chapman

Here are two enhancements for the Atari version of COMPUTE!*'s popular spreadsheed program* SpeedCalc *(published March 1986). These new features give you greater control over printed output and allow you to copy or move blocks of cells without recalculating the entire spreadsheet. A disk drive is required.*

Atari *SpeedCalc* is an excellent spreadsheet program, but even a good program can be improved here and there. "Enhancements For Atari *SpeedCalc*" makes several modifications to *SpeedCalc* to increase its power and convenience. Type in the program and save it to disk or tape, then run it.

When the program begins, you are prompted to insert a disk containing Atari *SpeedCalc. Make sure you have a backup copy of* SpeedCalc *stored safely on another disk, in case you experience a disk error or change your mind about using the enhanced version of* SpeedCalc. Press RETURN when the disk is in place. The enhancement program automatically appends the necessary code to the *SpeedCalc* AUTORUN.SYS file. After a few moments, the computer prints DONE. To enter *SpeedCalc*, remove or disable BASIC, then reboot the system.

Selective Printing

When printing to a device (a printer, disk drive, or the screen), the original *SpeedCalc* always starts printing at the upper left cell in the spreadsheet (cell AA1). This feature effectively limits the width of any printout to seven- or eight-cell columns on an 80-column printer. The enhanced version of *SpeedCalc* has the ability to send the contents of any block of cells to the device you select.

To print out a selected block of cells, move the cursor to the bottom right cell of the block that you want to print, then press CTRL-P (hold down CTRL, then press P). When prompted for the output device, enter P: to select the printer, E: to select the screen, or D: followed by a filename to print to a disk file. Now move the cursor to the top left cell of the block you wish to print, then press RETURN. *SpeedCalc* prints only the selected block.

Improved Move And Copy

The new version of *SpeedCalc* also has the ability to copy or move blocks of cells without recalculating. This permits you to piece together sections of the spreadsheet for printing without causing calculation errors. For example, you may want to move a column of titles just to the left of the cells to be printed. Recalculation during copy and move operations is now consistent with *SpeedCalc*'s automatic recalculation mode. If automatic recalculation is turned on, copy and move commands cause the entire spreadsheet to be recalculated. If automatic recalculation is turned off, copy and move simply move the contents of the selected block from one place to another within the sheet. Just as in the original version, you can toggle automatic recalculation mode on or off by pressing CTRL-R.

Enhancements For Atari *SpeedCalc*

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
NM 1Ø  REM PRINT ROUTINE ENHA
       NCEMENTS FOR SPEEOCALC
BI 2Ø  REM THIS PROGRAM APPEN
       OS SEVERAL PATCHES TO
       THE ORIGINAL SPEEOCALC
DB 3Ø  TRAP 43Ø
NI 4Ø  CHECKSUM=Ø:NBYTES=BØ
JL 5Ø  FOR BYTE=1 TO NBYTES:R
       EAO ABYTE:CHECKSUM=CHE
       CKBUM+ABYTE:NEXT BYTE
DJ 6Ø  IF CHECKSUM<>7369 THEN
        PRINT "ERROR IN OATA
       BTATEMENTS":GOTO 44Ø
CI 7Ø  OIM A$(1)
PN BØ  PRINT "{CLEAR}INSERT B
       PEEOCALC OISK & PRESS
       RETURN":INPUT A$
DD 9Ø  CLOSE #1
ID 1ØØ  OPEN #1,9,Ø,"O:AUTORU
        N.SYS":REM APPENO PAT
        CHES TO ENO OF ORIGIN
        AL FILE
EO 11Ø  RESTORE 17Ø
BC 12Ø  PRINT "WRITING..."
DF 13Ø  FOR BYTE=1 TO NBYTES:
        REAO ABYTE:PUT #1,ABY
        TE:NEXT BYTE
FP 14Ø  CLOSE #1
JO 15Ø  PRINT "OONE":ENO
CB 16Ø  REM $1FØØ-$1F2B, 1ST
        PATCH
HC 17Ø  OATA Ø,31
KK 1BØ  OATA 43,31
DD 19Ø  OATA 162,Ø,32,199,5B,
        32
BI 2ØØ  OATA BB,46,173,17,66,
        2Ø5
CD 21Ø  OATA 1,66,144,24Ø,173
        ,1
JF 22Ø  OATA 66,133,2Ø5,173,1
        9,66
JE 23Ø  OATA 2Ø5,2,66,144,227
        ,169
CN 24Ø  OATA 65,16Ø,79,162,Ø,
        32
HC 25Ø  OATA 199,5B,32,B9,33,
        162
ID 26Ø  OATA 4,96
HN 27Ø  REM
BK 2BØ  REM $1F4Ø-$1F4A, 2NO
        PATCH
KP 29Ø  OATA 64,31
KI 3ØØ  OATA 74,31
MC 31Ø  DATA 173,143,62,24Ø,3
FP 32Ø  OATA 76,15Ø,51,76,152
        ,33
HK 33Ø  REM
CH 34Ø  REM $2COB-$2CEØ,REPL
        6 BYTES IN SPEEOCALC
        COOE
OC 35Ø  OATA 219,44
NP 36Ø  OATA 224,44
PD 37Ø  OATA 32,Ø,31,32,199,5
        B
HP 3BØ  REM
PL 39Ø  REM $31C9-$31CB,COPY/
        MOVE PATCH
NK 4ØØ  OATA 2Ø1,49
NN 41Ø  OATA 2Ø3,49
ED 42Ø  OATA 76,64,31
LI 43Ø  ERR=PEEK(195):PRINT "
        ERROR- ";ERR
KD 44Ø  PRINT "PROGRAM ABORTE
        O!"
8D 45Ø  CLOBE #1                  ©
```

Commodore 128 Machine Language

Part 2

Jim Butterfield, Associate Editor

This second in a series of articles on programming the 128 computer in its 128 mode, explores the built-in machine language monitor and looks at ways to link machine language programs to BASIC.

A Monitor At Your Fingertips

Some of the earlier Commodore products had no built-in machine language monitor. To work on machine language on the VIC-20 or Commodore 64, for example, you had to load a machine language monitor from tape or disk, or rely on a plug-in cartridge. Other products had simple monitors: Many PET/CBM models had monitors which could display and change memory, save or load programs, and not much else. The built-in monitor on the Commodore 128 has many attractive features; the best way to learn them is to try them.

Type MONITOR and press RETURN. You'll see the familiar register display, with values under the titles: PC (program counter), SR (status register), AC (accumulator or A register), XR (X register), YR (Y register), and SP (stack pointer). They are all similar to what you may have met on other machines except that the value under PC looks a little odd. It has five digits instead of four. The extra digit at the beginning is the *bank number*, and since it's an *F*, we're in bank 15.

We've noted previously that *bank* isn't quite the right term. We should more properly say *configuration* 15, since each configuration consists of a mixture of memory elements. Figures 1 and 2 show the configurations for banks 15 (the default) and 0. You'll notice that for addresses below $4000, both bank 0 and bank 15 use exactly the same memory. Thus, the contents of address $F1000 is exactly the same as the contents of address $01000. In fact, it's the same memory. We'll look for ourselves in a few moments.

Number Conversion

You may be quite comfortable with hexadecimal numbers. You may even be able to do hex-to-decimal conversions in your head and amaze your friends. I can't, however, and I like the number conversion features that are built into the monitor.

Figure 1: Bank 15

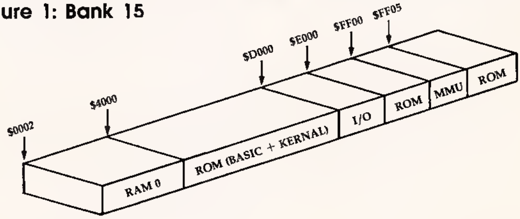
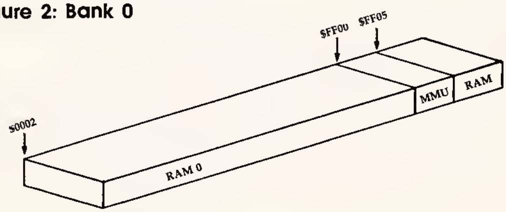


Figure 2: Bank 0



Only NRI teaches you to service and repair all computers as you build your own 16-bit IBM-compatible micro

Now that computers are firmly established in offices—and in homes, too—the demand for trained computer service technicians surges forward. The Department of Labor estimates that computer service jobs will actually *double* in the next ten years—a faster growth rate than any other occupation.

Total systems training

No computer stands alone . . . it's part of a total system. And if you want to learn to service and repair computers, you have to understand computer *systems*. Only NRI includes a powerful computer system as part of your training, centered around the IBM-compatible Sanyo 550 Series computer.

As part of your training, you'll build this highly rated, 16-bit IBM compatible computer system, assemble Sanyo's "intelligent" keyboard, install the power supply and disk drive, interface the high-resolution monitor and dot matrix printer, even expand the memory from 128K to 256K RAM. It's confidence-building, real-world experience that includes training in programming, circuit design, and peripheral maintenance.

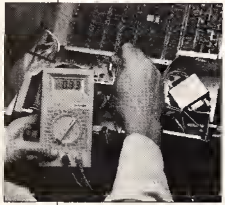
No experience necessary— NRI builds it in

Even if you've never had any previous training in electronics, you can succeed with NRI training. You'll start with the basics, then rapidly build on them to master such concepts as digital logic, microprocessor design, and computer memory. You'll build and test advanced electronic circuits using the exclusive NRI Discovery Lab®, professional digital multimeter, and logic probe. Like your computer system, they're all yours to keep as part of your training. You even get over $1,000 worth of software, including the popular WordStar and CalcStar.

Send for 100-page free catalog

Send the coupon today for NRI's 100-page full color catalog, with all the facts about computer training. Read detailed descriptions of each lesson, each experiment you perform. See each piece of hands-on equipment you'll work with and keep. And check out NRI training in other high-tech fields such as Robotics, Data Communications, TV/Audio/Video Servicing, and more.

Mail the coupon today, and see how NRI can prepare you for advancement and new careers in the exciting world of electronics. If the coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, DC 20016.



NRI is the only home study school that trains you on a total computer system. You'll install and check keyboard, power supply, disk drive, printer, and monitor, following step-by-step directions.

Your NRI total systems training includes all of this • NRI Discovery Lab® to design and modify circuits • Four-function digital multimeter with audio cassette training • Digital logic probe for visual examination of computer circuits • Sanyo 550 Series computer with "intelligent" keyboard and 360K double-density, double-sided disk drive • High-resolution monochrome monitor • RAM expansion module to give you powerful 256K memory • 120 CPS dot matrix printer with near-letter-quality mode • Easy Writer I, WordStar, CalcStar bundled software • Reference manuals, schematics, and bite-sized lessons.

SEND COUPON TODAY FOR FREE NRI CATALOG!

NRI SCHOOLS

McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, NW, Washington, DC 20016
We'll give you tomorrow.

For Career courses
approved under GI Bill
□ check for details

☑ **CHECK ONE FREE CATALOG ONLY**

- □ **Computer Electronics with Microcomputers**
- □ TV/Audio/Video Servicing
- □ Satellite Electronics
- □ Robotics & Industrial Controls
- □ Data Communications
- □ Industrial Electronics
- □ Communication Electronics
- □ Electronic Design Technology
- □ Telephone Servicing
- □ Digital Electronics Servicing
- □ Basic Electronics
- □ Bookkeeping & Accounting
- □ Appliance Servicing
- □ Small Engine Repair
- □ Air Conditioning, Heating, & Refrigeration
- □ Locksmithing & Electronic Security
- □ Building Construction
- □ Automotive Servicing

Name (Please print) ____________________ Age ______

Street ____________________

City/State/Zip ____________________

Accredited by the National Home Study Council

198-096

We've talked about hexadecimal address $4000 already. Let's find its value in decimal.

Type in the value $4000 on a line by itself and press RETURN. You'll see a display of this number as it appears in various number bases. First, the hexadecimal number., The dollar sign means hex, of course, so the monitor simply echoes what you typed in: $4000. The next line starts with a plus sign (+). To the 128's monitor, the plus sign means decimal. So you can see that $4000 equals decimal 16384. The following line starts with an ampersand (&), which means *octal*, a notation that's rarely if ever used with Commodore machines. (Octal numbers are base 8, so &40000 is equal to four times eight raised to the fourth power.) Finally, the number that starts with a percent sign (%) is the *binary* representation of $4000. Since the computer's internal code is always binary—not decimal or hexadecimal—it's sometimes useful to be able to look at a number this way.

You may also convert a decimal number to the other bases by typing it in, leading off with a plus sign. If you like, try entering +16384 and watch the computer figure out that it's the same as $4000. And if you ever need to do so, you can convert from octal or binary the same way.

Conversions are convenient, but the monitor includes another bonus: *Any number may be entered in any base, any time.* If you put in a number without a prefix, the monitor will assume you mean it to be hexadecimal. But you can slip in a decimal number anywhere by prefixing it with the plus sign. We'll be doing this; you'll see how handy it is.

Looking At Memory

You may display memory with the command M. If you follow M with two addresses, the monitor displays all the values between them. Thus, to display the contents of addresses $1000–$1029, just type M 1000 1029 and press RETURN.

You'll get more than you bargained for. Depending on whether you are on a 40-column or 80-column screen, the monitor will display 8 or 16 memory locations at a time. Each group of locations is on a single line, with the address of the first item on the line showing at the left. We asked for 42 locations, but we got 48, since the computer always finishes the line it's working on.

On the right, we see the ASCII character equivalent of the contents of the memory locations; some locations don't happen to have an alphanumeric equivalent, in which case a period is printed. If you display the addresses suggested above, you'll see some readable text in this area. The zone of memory we're looking at holds the function key definitions.

Just to confirm something that was said before, try using M to display memory locations F1000–F1029. That's bank 15 instead of bank 0, but you'll see that it is in fact the same memory. And you might like to try M +4096 +4137 which uses decimal addresses for the same locations.

If you follow an M command with only one address, you'll get a fixed number of memory locations. This can save you typing, and here's a tip for browsing through large amounts of memory: If you type M alone with no addresses, you'll get a continuation of the last memory display.

Making Changes Directly

The simplest way to change memory is to display the area you're interested in, then move the cursor back and type over the values on the screen. When you press RETURN, the monitor enters all the values for that line. It's a bit like screen editing in BASIC.

Try it. If you have displayed memory as suggested above, you may see the word GRAPHIC on the right-hand side of the memory display. Let's change the *G* stored in memory to a *T* so that it says TRAPHIC. The code for a *G* is $47; it's found in the left-hand part of that line. Move the cursor over the 47 and type 54, which is the code for *T*. Now press RETURN and the memory change is made.

Remember that you can't change the right-hand ASCII side of the display. And by the way, this is *not* the recommended way to change the function key definitions. It's easier (and better) to use BASIC's KEY command.

You can't change locations in read only memory (ROM). Try this: M F4200 F4200 will show you part of the BASIC ROM. Move the cursor back, type over a value, and press RETURN. You'll see from the display that the original values have been restored and ROM has not changed. Here's a note for technical types: The values from the line have "poked through" into the RAM memory which lies beneath ROM, but the monitor shows only the ROM.

The first character on the memory display line is the greater-than sign (>). This is in fact a synonym for the change memory command. On rare occasions, you might like to use this command directly.

Here's a typical case where the greater-than sign might be typed: You want to change a single location in an I/O chip. Using the "display and type over" method, you'd change 8 or 16 locations at a time. Usually, that's okay, but I/O chips are delicate and you don't want to change other registers accidentally. As a simple example, you might like to change the 40-column border color to red, but you don't want to change anything else. You may type >FD020 2 (remember that the I/O chips are in bank 15) and the border will change. The monitor will display a full line of memory locations, but you've changed only one. By the way, did you notice that the address you changed does not now contain the value 2 you put in? Funny things, I/O chips. If you're interested, you might type $D020 to ask the computer what decimal address in bank 15 you have changed. You might recognize the answer, +53280.

Write A Simple ML Program

Let's write a short program to print a line of asterisks. We'll use the built-in *assembler*. Here goes:

```
A 1500 LDX #0
```

The A means assemble. The address at which we will put this instruction is 1500; it's in hexadecimal (put a dollar sign in front if you like). The instruction itself is LDX #0, load counter X with a value (the # character means a value, not an address) of zero. Press RETURN and you'll see that the line has changed to

```
A 01500 A2 00 LDX #$00
```

The machine code in addresses 1500 and 1501 (bank 0, but in this area that's the same as bank 15) is hex A2 00. These two bytes have been placed in memory, and the monitor is ready for your next line of code; in fact, it has typed part of it for you. Complete the next line so that it reads

```
A 01502 LDA #$2A
```

This instruction, when the program runs, will load the ASCII code for an asterisk (hex 2A) into the A register; that's the register we use for printing. Continue with

```
A 01504 JSR $FFD2
A 01507 INX
A 01508 CPX #+20
```

The first instruction in this group prints a character, calling the Kernal ROM routine usually known as BSOUT (also known in the Commodore 64 as CHROUT). The next adds one to the X register, which we're using as a counter. The last instruction says, "Compare the counter with decimal 20." Note the plus sign for decimal. When you press RETURN, the line changes to

```
A 01508 E0 14 CPX #$14
```

The value 20 has been changed to hexadecimal. Don't be surprised; it's still the same number. Continue entering with

```
A 0150A BNE $1504
A 0150C LDA #$0D
A 0150E JMP $FFD2
```

The instruction BNE $1504 sends the program back to print again if we haven't reached 20 characters. The sequence LDA #$0D:JMP $FFD2 prints a carriage return and terminates the program (we know that the ROM routine at $FFD2 ends with RTS, so we can save a little code by using that RTS to return, rather than ending with the more conventional JSR $FFD2:RTS). After typing the last line, the computer prompts you with A 01511. Simply press RETURN to end the assembly.

If you like, you can proofread your program by entering the command D 1500 150C. The D command is for disassemble, which performs an activity more or less the reverse of an assembly.

Starting Up

You can go to this program with a G (go) command, which doesn't permit a return. Better, you can call it with a J (jump subroutine) command. But first, *you must think about what bank you are in.*

If you enter the command J 1500, you'll have a disaster on your hands. Why? Because you're entering bank 0 which contains no Kernal ROM and no I/O chips. Remember, the program uses the Kernal ROM routine BSOUT to print each character. If you JSR to this routine when the Kernal ROM is absent, you'll never print those asterisks, and your program will almost certainly fail. If you really want to call this program from the machine language monitor, invoke bank 15 with J F1500.

It's also quite simple to call the routine from BASIC. First, find the starting address. Type $1500 and read the answer, decimal +5376.

Back To BASIC

Return to BASIC by giving the X (exit) command. You'll see the familiar READY response of BASIC. Now type NEW (don't worry, your machine language program won't be harmed) and enter the following program:

```
100 BANK 15
110 SYS 5376
120 PRINT "THIS WORKS"
130 SYS 5376
140 PRINT "WITHOUT PROBLEMS"
150 SYS 5376
```

Run the program and you should see a row of asterisks. If you've done these exercises, you should have a feeling for the 128's machine language monitor. It's convenient and flexible. In upcoming articles we'll learn more about the monitor, and how to link BASIC and machine language programs together. ©

Stringing Atari Machine Language

Robert Martinsons

Storing Atari machine language in a string is a time-honored technique, but how do you get the ML into the string in the first place? This program does the job automatically, creating the necessary string and appending it to the BASIC program of your choice. It's easy and very fast.

A good way to enhance the performance of BASIC programs is to use machine language subroutines for tasks which either take too much time or consume too much memory. And one of the most popular places to store short ML routines is in an Atari BASIC string. Once the ML code is stored in a string, BASIC's ADR function can calculate the string's address, and the USR function can call it.

Short machine language routines can be dealt with by manually typing them into strings, but this can be somewhat tricky, since it usually involves typing strange-looking control characters. Another possibility is to use DATA statements which BASIC can READ under program control. Neither of these methods is attractive for large routines, however. Substantial ML programs are usually written with an editor/assembler, which produces a binary file as output. The problem, then, is how to convert the contents of a binary file into a string that BASIC can easily handle.

The routine that accompanies this article solves the problem of converting binary files into string form. It reads binary data from a disk or tape file, stores it in a series of strings through the editor's *forced read* mode, then deletes itself from memory. Type in the program lines listed below, then LIST the routine to disk or tape. Do not save the routine: It must be LISTed so that you can later ENTER it into memory without disturbing a program that's already present.

Stringing Along

To use the routine, first load the BASIC program to which you would like to add a machine language routine. Of course, the ML routine is one which normally resides in a binary file. (Note that the ML routine *must* be relocatable, since Atari BASIC strings can move around in memory while a program runs.) The BASIC program must not use any line numbers higher than 31499, since this routine itself uses the lines beginning at 31500. Next, ENTER the routine from disk. This brings it into memory without altering the BASIC program. To activate the routine, type GOTO 31500 and press RETURN.

The program begins by requesting the filename of your binary file. Be sure to include the correct device prefix in your response. For instance, to read the binary file CODE.BIN from disk, enter D:CODE.BIN at the prompt. At the next prompt, enter the name of the BASIC string which will hold your machine code. Limit the name to eight characters or fewer (if you enter too many characters, the routine automatically truncates the name). Answer the last prompt with the line number where you want the new ML strings to begin. When answering this prompt, you should take care not to start the new lines at a place which would overwrite existing lines. A safe rule of thumb is to allow ten line numbers for every 256 bytes of machine language.

At this stage, the routine begins reading the ML code into memory and converting it into strings. When the process is complete, the routine deletes itself, leaving your original program plus the strings that contain the machine language. Before you can resave the program, you must manually add a DIMension statement for the new string and add USR calls for the routine where needed. It's also a good idea to LIST the revised program to disk, type NEW, and ENTER it again, before saving it a final time. In this way you can clear out all the variables used by the deleted routine.

The Editor Does All The Work

For those who are interested, here is a short explanation of how the

From the publishers of *COMPUTE!*

A Turning Point For Atari?
Report From The Winter Consumer Electronics Show
COMPUTE!
The Leading Magazine Of Home, Educational, And Recreational Computing
New Technologies: The Converging Digital Universe
Why Everything From Computers & CD-ROMs To TVs & Satellites Are On A Collision Course
Adding Power To ST BASIC
Valuable Techniques For Atari Programmers
Apple Hi-Res Graphics
Mix, Flip, And Copy High-Resolution Screens
64 Screen Genie
A Magic Method For Character Graphics
AmigaDOS Batch Files
Take Control Of Your Amiga
Atari FontMaker
Design Your Own Custom Characters
COMPUTE! DISK

September 1986 *COMPUTE!* Disk

All the exciting programs from the past three issues of *COMPUTE!* are on one timesaving, error-free, floppy disk that is ready to load on your Atari 400/800, XL, and XE. The September 1986 *COMPUTE! Disks* contain the entertaining and useful Atari programs from the July, August, and September 1986 issues of *COMPUTE!*.

The September 1986 *COMPUTE! Disk* costs $12.95 plus $2.00 shipping and handling and is available only from COMPUTE! Publications.

For added savings and convenience, you may also subscribe to the *COMPUTE! Disk.* At a cost of only $39.95 a year (a $12.00 savings), you'll receive four disks, one every three months. Each disk will contain all the programs for your machine from the previous three issues of *COMPUTE!*. To order a subscription, call toll free 800-247-5470 (in IA 800-532-1272).

This is an excellent way to build your software library while you enjoy the quality programs from *COMPUTE!*.

Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

For more information or to order individual issues of the September 1986 *COMPUTE! Disk,* call toll free 1-800-346-6767 (in NY 212-887-8525) or write *COMPUTE! Disk*, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE!™ Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE!, COMPUTE!'s Gazette, COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

conversion routine works. All Atari binary files have a six-byte header, which contains the information shown in the table.

Typical Binary File Header

Byte	Number	Number	Description
1	255	FF	Identification
2	255	FF	code for binary load file
3	0	00	Starting address (LSB)
4	10	0A	(MSB)
5	72	4C	Ending address (LSB)
6	13	0D	(MSB)

The conversion routine opens the binary file and executes a CIO (Central Input/Output) system call to bring in the first six bytes. It examines these and confirms that you have accessed a binary file, and then computes the file size by subtracting the starting address from the ending address. Next, a subroutine which dimensions a temporary string (TEMP$) is created and executed. For the sample header shown, the dimension of TEMP$ will be 841. TEMP$ becomes the input buffer for the next CIO call which reads in the remainder of the binary file.

A loop beginning at line 31610 now begins to break the data from TEMP$ into segments short enough to be stored in a BASIC line. Each new string will hold 90 bytes unless we find the ATASCII equivalent of a quotation mark (34) or carriage return (155). These values are handled separately to avoid confusing the screen editor.

The POKEs in the subsequent lines switch the editor into forced read mode, causing it to enter the new line just as if you'd typed it manually and pressed RETURN. Because the address of TEMP$ moves every time the editor enters a new line, its address is recomputed at the beginning of each loop. After the last byte of data has been packed into the new string, the conversion routine again uses forced read mode to delete itself from the finished program.

Chances are that you've been using a more manual method of embedding your assembly language routines into BASIC. If so, this routine should become a welcome part of your toolkit. Sit back and enjoy watching the screen editor do all the work. A final note: Every effort was made to keep the program as compact as possible. Therefore, no REMark statements are included and error trapping is held to a minimum.

String Atari Machine Language

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
BD 31500 CLR :GRAPHICS Ø:IND
         EX=1:LINENO=Ø:STRTL
         INE=45:DIM BUFFER$(
         8),FILNAM$(15),STRN
         AME$(8),A$(1):CIO=A
         DR("h▥{P}LV▤")
GN 31510 ? "Enter filename f
         or binary load file
         ":INPUT FILNAM$
BO 31520 ? "Enter BASIC stri
         ng name":INPUT STRN
         AME$
NL 31530 ? "Enter starting l
         ineno for string":I
         NPUT LINENO
PK 31540 A=ADR(BUFFER$):OPEN
          #1,4,Ø,FILNAM$:POK
         E 85Ø,7:B=INT(A/256
         ):POKE 852,A-256*B:
         POKE 853,B:POKE 857
         ,Ø
CO 31550 POKE 856,6:N=USR(CI
         O):IF PEEK(A)<>255
         OR PEEK(A+1)<>255 T
         HEN CLOSE #1:? "ERR
         OR: Not a binary fi
         le":STOP
HC 31560 FILSIZ=(PEEK(A+4)+2
         56*PEEK(A+5))-(PEEK
         (A+2)+256*PEEK(A+3)
         )+1
PH 31570 GRAPHICS Ø:POSITION
          2,4:PRINT "3175Ø D
         IM TEMP$(";FILSIZ;"
         ):RETURN"
FK 31580 PRINT "CONT":POSITI
         ON 2,Ø:POKE 842,13:
         STOP
FK 31590 POKE 842,12:GOSUB 3
         175Ø:TEMP$(1)=" ":T
         EMP$(FILSIZ)=" ":TE
         MP$(2)=TEMP$:ADDRES
         S=ADR(TEMP$):B=INT(
         ADDRESS/256)
OC 31600 POKE 852,ADDRESS-25
         6*B:POKE 853,B:B=IN
         T(FILSIZ/256):POKE
         856,FILSIZ-256*B:PO
         KE 857,B:N=USR(CIO)
         :CLOSE #1
FB 31610 GRAPHICS Ø:ADDRESS=
         ADR(TEMP$):POSITION
          2,4:LINELIM=INDEX+
         89
OL 31620 IF LINELIM>FILSIZ T
         HEN LINELIM=FILSIZ
OM 31630 A$=TEMP$(INDEX,INDE
         X):IF A$=CHR$(34) O
         R A$=CHR$(155) THEN
          3169Ø
CA 31640 LINESTRT=INDEX:FOR
         INDEX=LINESTRT TO L
         INELIM
CM 31650 A$=TEMP$(INDEX,INDE
         X):IF A$=CHR$(34) O
         R A$=CHR$(155) THEN
          LINENO=INDEX-1:GOT
         O 3167Ø
PC 31660 NEXT INDEX:LINENO=L
         INELIM
HK 31670 PRINT LINENO;" ";ST
         RNAME$;"$(";LINESTR
         T;",";LINENO;")=";C
         HR$(34);
FK 31680 FOR I=LINESTRT TO L
         INENO:? "{ESC}";TEM
         P$(I,I);:NEXT I:? C
         HR$(34):GOTO 317ØØ
BM 31690 ? LINENO;" ";STRNAM
         E$;"$(";INDEX;",";I
         NDEX;")=CHR$(";ASC(
         A$);")":INDEX=INDEX
         +1
LB 31700 LINENO=LINENO+1:PRI
         NT "CONT":POSITION
         2,Ø:POKE 842,13:STO
         P
BC 31710 POKE 842,12:IF LINE
         LIM<FILSIZ THEN 316
         1Ø
BP 31720 GRAPHICS Ø:POSITION
          2,4:FOR I=3149Ø TO
          3165Ø STEP 1Ø:? I:
         NEXT I:? "CONT":POS
         ITION 2,Ø:POKE 842,
         13:STOP
FB 31730 POKE 842,12:GRAPHIC
         S Ø:POSITION 2,4
OB 31740 FOR I=I TO 3175Ø ST
         EP 1Ø:? I:NEXT I:?
         "POKE 842,12":POSIT
         ION 2,Ø:POKE 842,13
         :STOP
```

©

The World Inside the Computer

Fred D'Ignazio, Associate Editor

Sandbox Fred And His Media Maniacs

Recently, while I was in Vancouver, Canada, at the World Congress on Education and Technology, I was asked to teach an intensive week-long teacher's workshop at Simon Fraser University, one of Canada's leading universities. The first night of my course at Simon Fraser, I learned that most of the teachers taking my course were novices in electronic media, and that some of them had never even touched a computer. They saw me as a media expert and hoped the course would give them some hands-on experience creating teaching units with different media equipment.

The learning resources center where I taught the course has one of the richest collections of electronic media that I have ever seen. To be frank, there were so many darkrooms, multitrack tape decks, audio/video mixers, computers, projectors, and the like, that it was downright intimidating. Even I was scared, so how were my fearful teachers to acquire the courage to use all that stuff?

Electronic Sandbox

As I stood in front of my class that first night, I dug deep inside myself for the one thing that I stood for, the one thing that would charge up the class to leap into the media with gusto and pizzazz. Then I thought of the magic word: *sandbox*. To me a sandbox is more than four boards and a bag of sand. It is a metaphor for play, storytelling, world building, and for a child's personal journey of exploration and discovery. And sand is a metaphor for what good media should be—rich, malleable, and gritty. Playing with media should be a multisensory experience. As with sand, you should smell it, taste it, and touch it. It should get in your ears, in your shorts, and in your hair.

I told my teachers that I was not a media expert nor a teacher, but an author. And what I could bring to the course was not technical expertise, but my imagination, my gift for storytelling, and my playfulness. I wasn't going to teach them. I was going to climb into the sandbox with them as "head kid." This approach was not what the teachers expected, but it turned out to be just what they needed.

We began the week with imagination exercises: We closed our eyes and tried to imagine holding a baby. We tried to smell the baby, touch the baby, taste the baby, see the baby, and hear the baby coo, laugh, and cry. We explored how media affects the imagination and how imagination is instrumental in creating good media. Although many of the students had never used a computer before, some had, and the veterans coached the beginners so they could sign on to the university's network. Beginning that first night we kept an electronic journal online that eventually amounted to 50 typed pages. We used the journal to reflect on the week's experiences and to examine the effectiveness of the sandbox approach to learning electronic media.

The teachers eventually divided themselves, according to their interests, into four groups:

- Mandalas (video, animation, sound synthesis, poetry, the arts)
- Choclit (a cartoon with sound synthesis)
- The Sandbox Saga (desktop publishing)
- The Media Maniacs (a documentary video of our week together)

Although no one had planned it, all the groups became intensely involved in storytelling and the imagination. And the groups divided neatly into Mandalas and Choclit, which were an exercise of the imagination looking outward, and Sandbox Saga and Media Maniacs, which showed the imagination looking inward at ourselves. The Media Maniacs theme came from the *Fred's Media Maniacs* buttons that one of the teachers made for us with the help of his mentally retarded students.

Jumping In Headfirst

By week's end I knew that grown-up, high-tech sandboxing can really work. Teachers threw themselves into their projects with ferocious energy and creativity. They mastered machines that they had never even seen before, fussed with buggy software and malfunctioning equipment, and moved on. Nothing stopped them. And their movies, stories, and cartoons were delightful.

But sandboxes have their dark side, too, and we stumbled into this area often. Playing is good, but sometimes there is nothing in a sandbox to play with. My metaphor of a sandbox as a free, unstructured environment encouraged the teachers to be childlike and playful, but they needed guidance and instruction to produce real results. "It's exciting to watch people playing in a sandbox," said one of my students. "But it's no fun at all if you can't get in."

The best part came at week's end when we held a Sandbox Media Festival for a class of computer software teachers. All the teachers' products were terrific, but I especially liked the ones done by the Media Maniacs. One of its producers, Morey, had gotten his three-year-old son, Cameron, to play the part of Sandbox Fred as a child. In the sequence Cameron zigs and zags around the sandbox in his red shorts and a white sun hat and says, "I'm Sandbox Fred, and I like to play in sandboxes. I'm Sandbox Fred, and I like computers. I'm Sandbox Fred, and I have to go potty on the tree."

Computers and Society

David D. Thornburg, Associate Editor

Sampled Sounds

While the debate continues to rage over the destiny of the home computer, specialized programmable computers are showing up in people's homes in record numbers. These computers are the inexpensive music synthesizers manufactured by Casio, Yamaha, Seiko, Kawai, and several others.

In an earlier column I mentioned that the acceptance of the MIDI interface standard has resulted in a powerful merging of synthesizer technology with personal computers. I expect that within a few years every new personal computer will have a built-in MIDI interface.

Music For Everyone

Our love affair with music is extraordinary. At any time of the day or night you can turn on your radio and find that the vast majority of stations are playing music. Given the popularity of recorded music and concerts, you might conclude that we are more interested in hearing music than making it. While this is probably true to some extent, it's not as pervasive as it seems. Musical instruments sell briskly.

Millions of people want to enjoy music by playing it themselves. Historically there have been two barriers to this creative urge. The first is the difficulty of learning to play a conventional musical instrument, and the second is the difficulty of learning to read and write music using traditional notational schemes. Faced with the need to practice for years, many would-be musicians give up in frustration.

From the moment it is brought home, the modern digital synthesizer allows music to be created. Unlike a real trumpet, whose first sounds seem better suited for burglar alarms than for music, a synthesized trumpet sounds sweet from the very beginning.

In addition to providing high-quality sounds, the inexpensive modern synthesizer provides additional help to musicians in the form of sophisticated rhythm sections, automatic arpeggios and chords, and even the ability to sequence several tracks of music into a completely orchestrated piece. All these features can be found at the local discount store for under $200.

New Instruments

If I felt for a moment that synthesized instruments were going to replace traditional instruments, I would be concerned. Instead, we are seeing the synthesizer emerge as a class of instrument in its own right, taking its place next to traditional instruments.

The most exciting aspect of synthesizers is that they can produce sounds unavailable in traditional instruments. If you think about it, musical sounds are made in one of four ways: by hitting something (drums or pianos), plucking something (harpsichords, guitars), blowing air into or across something (organs, horns), or scratching two things together (violins). The synthesizer can emulate many of these sounds, but more importantly, it can be used to create sounds that can't be made by traditional methods. This allows the design and creation of new musical instruments by a new breed of craftsperson—one who works with programs rather than with chisels and glue.

The SK-1

If there is a major limitation to modern synthesizers, it is that new sounds can be hard to implement. For instance, the Yamaha DX-7, one of the standard instruments in the field, is difficult to program without the use of a separate computer.

A recent entry into the low-cost synthesizer market has made this task a lot easier. This instrument is Casio's SK-1 sampling keyboard, which retails for well under $200. The computer in the instrument allows sounds to be captured from external sources through a built-in microphone. Suppose you would like to make an instrument that sounds like a hammer hitting a pipe. To capture this sound, you need only place the SK-1 near a pipe (an external mike can be used), press the Sample key on the synthesizer, and hit the pipe with a hammer. The internal computer samples the sound for 1.4 seconds, encodes the sound digitally, and stores it in about 14K bytes of RAM. The sound you record is assigned to the A key. Once the sound is entered, you can play it at any pitch by pressing the appropriate key on the keyboard. You can also modify the sound's envelope after it is recorded.

Experimentation

The most exciting aspect of this instrument, and others like it, is that it stimulates creative experimentation. If it took hours to create new sounds, you might be reluctant to try offbeat ideas, simply because they might turn out to be a waste of time. With the SK-1, a new sound can be captured in a few seconds. As a result, new owners of the instrument typically spend the first day or so capturing everything from motorcycle engines to recited poetry and using these sounds to create new music.

This playful aspect of the synthesizer is its greatest strength. The computer in this synthesizer is completely transparent to the user. There is no barrier between your goal—music making—and a satisfying result. Technology has receded into the background to facilitate the creation of music, and another computer has quietly entered the home.

The Beginners Page

Tom R. Halfhill, Editor

That Other Computer Language

Usually when someone talks about a "computer language," we think of programming languages like BASIC, Pascal, Forth, Logo, and so on. These languages are of interest only to programmers—if you merely want to *use* a computer, you don't have to learn anything about these languages at all.

But no matter how far removed you want to remain from the inner workings of the machine, there is one computer language you *do* have to learn: *lingo*, all those complicated terms and odd slang words that only computer experts seem to understand. You know what I mean: "Oh, you're having RS-232 glitches? This is just a kludge, but try checking your DTR pin and changing duplexes, and if that doesn't work, flip your floppy and warmboot DOS with an ASCII batch file."

Alien Conversations

When you're a struggling computer-illiterate, it's tempting to assume that this kind of gibberish was invented merely to exclude outsiders from the inner circle. Actually, every occupation, hobby, and field of interest has its own lingo. Listen to yourself someday when talking to a co-worker or a fellow student; you'll be surprised how alien the conversation might sound to someone who is uninitiated.

This was brought home to me recently when I was helping a new computer owner learn to set up and use his system. Suddenly he interrupted: "*Boot it up?* Does that mean the same thing as *turn it on?*" I was caught off-guard. Once you learn lingo, it's amazing how fast you take it for granted.

To help clear up any similar confusion you may be experiencing, let's take a look at some of the terms which make up computer lingo:

Back door A secret method of gaining entry to a restricted program by circumventing the password protection. Usually planted by the programmer.

Boot To start up a computer system, usually by switching on the power. Some computers equipped with disk drives must be booted with a disk in the drive (a *boot disk*) that contains the disk operating system (DOS). Commodore computers are exceptions, because DOS is built into the drives themselves. On the Amiga and early versions of the Atari ST, the computer's operating system itself must be loaded from disk when booting.

Bug A malfunction of hardware or software that can often be replicated. Usually the fault of the programmer or designer.

Bus A connector on a computer into which accessories and cables are plugged. Usually referred to as a *system bus* or *expansion bus*.

Clone A computer that is designed to run the same programs and accept the same accessories as another computer made by a rival manufacturer. Clones typically sell for less than the computer they're imitating. The computers most often cloned are the IBM PC and Apple II.

Cold start To boot up a computer system by switching on the power.

Crash Sudden, total failure of a program or computer system. The program or computer refuses to acknowledge commands, usually because of a bug or glitch.

Daisychain Two or more accessories—such as disk drives, a printer, or a modem—all hooked together sequentially to form a chain. The term can also be used as a verb to describe the process of connecting a device to the chain.

Elegant Perhaps the highest compliment that can be paid to the design of a program or piece of computer hardware. A solution that achieves both success and efficiency.

Gender changer An adapter that turns a male plug into a female jack or vice versa. Intended for matching cables to various kinds of computers and accessories.

Glitch A momentary malfunction of hardware or software. Similar to a bug, but more transitory, and not necessarily the fault of the designer or programmer.

Hacker Originally, someone who became deeply absorbed in programming or exploring the innards of the machine, even if nothing practical ever resulted—sometimes to the point of obsession. Recently this term has taken on a different connotation, due largely to misuse in popular media. In this usage, a hacker is someone who gains access to a computer system with mischievous intent, often via a telephone link.

Kludge (Pronounced *klooj*) A sloppy design or an inelegant solution to a problem. It works, but is clumsy or inefficient.

Lockup The keyboard refuses to respond to typed commands. Usually indicates a crash.

Meg Short for *megabyte*, a measurement of computer memory capacity. One megabyte equals 1024 *kilobytes* (1024K). A kilobyte equals 1024 *bytes*. A byte, in turn, is roughly equivalent to one character of storage. Thus, a meg of memory can hold 1,048,576 (1024 × 1024) characters.

Motherboard The main circuit board inside a computer.

Warm start To reboot a computer system that has already been cold-started, but has crashed or needs to be reset for some other reason. Most computers have a reset button or special key sequence for this purpose. ©

IBM Personal Computing

Donald B. Trivette

Photo Labeling

There should be a law requiring all photographs to be labeled with the date and content; otherwise, how is one to remember when and where each snapshot was taken? Unfortunately, writing on the back of a photograph is about as much fun as writing on wax paper. Writing on a word processor, on the other hand, is lots of fun—so if we could somehow get our PC to print on the backs of photographs, we just might have something useful. The solution is the BASIC program listed below to print address labels, which stick nicely to almost any surface—including wax paper and photographs. In addition, the program incorporates features to print multiple labels with the same information and to date each label automatically.

The program reads a file named LABELS, which you create using a word processing program or text editor. The file must be in ASCII format, and the length of each line should not exceed the width of a label. The program is designed to use 3-1/2 × 15/16 inch, fanfolded, pressure-sensitive labels that may be purchased in most office-supply stores for about $7 per thousand. This size label holds five 34-character lines of text.

In order to separate one label from another, the program looks for a dash (–) in the first column of the data. If there is a number immediately after the dash, the program will print that many labels with the text that follows. The first line in the file must either be a blank or contain a date that will be appended automatically to each label. The following figure shows an example of a LABELS file.

```
 (July '86)
-15
Vacation at Yellowstone
-
Uncle Eric
-6
Family Reunion
Miller Park
Mayberry, N.C.
-
Joe and Phyllis
-
Aunt Mary's house
```

This file prints 15 labels for the photos taken at Yellowstone, 1 label for Uncle Eric's photo, 6 to be stuck on the backs of the reunion photos, and 1 each for Joe and Phyllis and Aunt Mary's house. The program prints only five lines to a label; lines after the fifth are discarded, but it's up to you to format the length of each line to stay within the label boundary. The program includes a line-up routine to make it easy to get the labels positioned in the printer.

Photo Labeler

```
GA IØ REM
AF 2Ø REM Program to print 3 1/2
       x 15/16 inch
NJ 3Ø REM Iabels for the backs o
      f photographs.
DD 4Ø REM First line in LABEL5 f
      ile may either be
OI 5Ø REM blank or a date. The -
       sign signals
CI 6Ø REM the end of one Iabel a
      nd beginning of
DP 7Ø REM a new one. The -n opti
      on may be used to
BN BØ REM print "n" identical la
      bels. Each label
DN 9Ø REM may have a maximum of
      34 characters
MA 1ØØ REM by 5 Iines.
PG 11Ø REM
GI 12Ø KEY OFF:CL5:DIM 5$(2Ø)
FB 13Ø X=1:I=1:5W=Ø:CNT=Ø
NO I4Ø OPEN "Iabels" FOR INPUT A
       5 #1
LJ 15Ø LINE INPUT #1,DAT$
BI I6Ø REM Ready printer and aIi
       gn IabeIs
DJ 17Ø REM Print a test IabeI.
IP 1BØ PRINT "Insert labels in p
       rinter and press"
OD 19Ø PRINT "any key to continu
       e..."
KO 2ØØ A$=INKEY$:IF A$="" THEN 2
       ØØ
DM 21Ø LPRINT "<<****";5PACE$(6)
       ;"Top Line";SPACE$(6);"**
       **>>"
PN 22Ø FOR I=I TO 5:LPRINT:NEXT
       I
KF 23Ø PRINT "Is Iabel aligned?
       (Y/N)"
CO 24Ø A$=INKEY$:IF A$="" THEN 2
       4Ø
BJ 25Ø IF A$="Y" OR A$="y" THEN
       27Ø ELSE 21Ø
LL 26Ø REM Read data from file
KG 27Ø IF MID$(A$,I,1)="-" THEN
       X=ABS(VAL(A$))
IN 2BØ IF X=Ø THEN X=1
BC 29Ø LINE INPUT #1, B$
PG 3ØØ IF MID$(B$,1,I)="-" THEN
       GOSUB 36Ø:A$=B$:I=I:GOTO
       27Ø
NF 31Ø 5$(I)=B$
JC 32Ø I=I+I
IE 33Ø IF EOF(1) THEN GO5UB 36Ø:
       PRINT:PRINT CNT;" Labels
       printed":END
HD 34Ø GOTO 29Ø
NN 35Ø REM Print LabeI(s)
GC 36Ø IF SW=Ø THEN 5W=1:RETURN
LN 37Ø I=I-I
JO 3BØ IF I>5 THEN I=5
NI 39Ø FOR J=1 TO X
IP 4ØØ CNT=CNT+I
DB 41Ø FOR K=1 TO I-1
JB 42Ø PRINT S$(K)
NP 43Ø LPRINT 5$(K)
OG 44Ø NEXT K
FN 45Ø PRINT 5$(I);DAT$
MB 46Ø LPRINT S$(I);DAT$
LG 47Ø FOR L=1 TO 5-I
NB 4BØ PRINT 5PACE$(4)
AB 49Ø LPRINT 5PACE$(4)
PJ 5ØØ NEXT L
DJ 51Ø PRINT  5PACE$(4)
PE 52Ø LPRINT 5PACE$(4)
OL 53Ø NEXT J
NJ 54Ø FOR K=1 TO I
FH 55Ø 5$(K)=5PACE$(4)
PL 56Ø NEXT K
NN 57Ø RETURN
QP 5BØ REM End of Labels Program
```

©

Telecomputing Today

Arlan R. Levitan

A Well-Deserved Feast

What restaurant in your home town has the best Szechuan fare? How about barbecue, or Mexican, Thai, or Continental cuisine? Make a mental list of those places, then figuratively fold it up and put it aside for a few moments.

About a year ago, while cruising through the message section of a Chicago-based bulletin board, I ran across a message that caught my attention. It announced the opening of a new bulletin board in the Detroit area for IBM PC and PC-compatible computer owners. Dubbed "The Business Board," it was located in a nearby suburb. I was intrigued. While there were dozens of Atari-, Commodore-, and Apple-oriented BBSs in and around the Motor City, there had been a distinct paucity of PC-related boards. Prior to this time, I had been calling boards in other cities—not an economical practice when you count the long distance charges. A local PC BBS might open up new fields of interest as well as relieve my pocketbook.

As the modem dialed the new board's number, I purposely held down my expectations. Bulletin boards come and go. Most are started by well-intentioned folks who don't realize how much work is involved in maintaining and operating a BBS. The life expectancy of an average new board is about 30 to 60 days.

Two Deadly Errors

Why such a high mortality rate? There are two common, often fatal mistakes. Many a would-be SYSOP decides to run a board during hours when his or her computer is not otherwise in use. These moonlight boards are usually down more often than they're up. As the novelty wears thin, the neophyte SYSOP soon decides that taking the board up and down constantly is more bother than it's worth. An even more deadly mistake is attempting to use the same phone line for both voice and BBS communications.

As I logged onto "The Business Board," I was pleasantly surprised to see a nice introductory bulletin with slick graphics. Based on a dedicated Compaq portable with a 30-megabyte hard drive, the BizBoard (as it's called by users) has a download area containing over 1000 files. That's one of the most complete and up-to-date collections of "freeware" and public domain software that I've run across in years.

Dedicated Downloading

A quick electronic chat with SYSOP Rick Brenner revealed that the BizBoard's collection of files is the result of untold hours of downloading from a dozen or so of the nation's best bulletin boards. Apart from the phone charges, which are not insignificant, that sort of activity represents a very substantial investment in time.

Brenner started his board to facilitate the exchange of information among professionals who use computers in business. In keeping with this special focus, access to the board is limited. Membership is by registration only and costs $25 per year. You must also participate actively in the board's message traffic. Those whose sole interest is in downloading files are politely dropped from the rolls (and given a refund of their registration fees).

While the BizBoard's house rules may seem straight-laced to some, they have succeeded in fostering an unusually high degree of computer literacy and esprit de corps among BizBoard members. There's much humor to be found in the message bases and recently-added special interest forums, in addition to useful technical information, discussions of some of the more obtuse business applications of microcomputers, and accounts of member experiences with new products.

When Onliners Meet Offline

In February of this year, at my suggestion, the local BizBoard membership met for some offline conferencing at a local French bistro. Prior to the event, some new members had voiced concerns about holding their own in face-to-face communications with established technical heavies. To add to the interest, the suit-to-sandals ratio among the twenty-odd group members ran just about fifty/fifty. How did it go? The dinner meeting had been scheduled to run from 6:30 to 9:00 in the evening. We were finally ushered out the door at 2:00 the next morning. Since that auspicious beginning, bimonthly dinner meetings have become a BizBoard tradition.

It's been a year now since the BizBoard began. Since then, my favorite BBS has garnered about a hundred members, and survived several hard disk crashes, power failures, and even a fried motherboard. Most of the credit is due to its hard-working SYSOP.

There are hundreds of Rick Brenners across the land running bulletin board systems for telecomputing enthusiasts. Their labor of love goes largely unheralded. Have you got a local BBS in your area that deserves recognition? Unfold that piece of paper you stashed away mentally a few minutes ago. In my book, September is National SYSOP Month. Put off buying that new piece of software until next month. Instead, treat your local SYSOP to a gastronomic feast as rich as the one proffered to you via the telephone lines day after day. You'll both be better off for the experience. ©

ST Outlook

Philip I. Nelson, Assistant Editor

Pointer Potpourri

Welcome to "ST Outlook." Beginning this month, I'm taking over COMPUTE!'s Atari ST column from Bill Wilkinson, who had agreed to do the column on an interim basis. By way of an introduction, I'm an ST owner and programmer, as well as a writer and editor. In addition to *COMPUTE!'s ST Programmer's Guide*, which I coauthored, I'm currently collaborating with COMPUTE! programmer Tim Victor on an upcoming book, *Mapping the Atari ST*, the first volume of which is scheduled for an early 1987 release.

Pick Your Pointer

Every ST owner is familiar with the way the mouse pointer changes appearance in response to system events. When you open an application from the desktop, or load a program from BASIC, the pointer changes from an arrow to a busy bee, and so on. In many situations, the ST manages the pointer shape automatically. But you can also change it under program control to suit your own needs.

This month's program shows how to access the ST's eight built-in pointer shapes from BASIC. It displays all the pointers in turn, prompting you to click the mouse button when you're ready to see the next one in the series. In addition to the familiar arrow and bee, you'll see two hand shapes, three different crosshair pointers, and a cursor shaped like a slender I-beam.

It's not difficult to see how alternate pointer shapes can come in handy. For instance, the bee does not automatically appear when you read or write to disk or perform other time-consuming chores in BASIC. While you can print the conventional PLEASE WAIT message under those circumstances, it's also prudent (and it adds a touch of elegance) to change the pointer to a bee. By reducing the user's temptation to fiddle with the menus or wave the pointer absent-mindedly, this little icon increases the chances that your program will work as intended. These cautions are doubly important because BASIC freezes program execution whenever the pointer is in motion and offers no easy means for disabling its own menus.

If you've used *1st Word*, the word processor supplied with the ST, you may recognize the pointing hand, which appears whenever you drag the pointer to define a block of text. The I-beam cursor, thin enough to fit neatly between text characters, is ideally suited to word processing and similar applications. The grabbing hand pointer is often used to manipulate objects such as window sliders. And the crosshairs are ideal for drawing or any activity that requires precise positioning.

Suit Yourself

Of course, you're free to use these pointers as you please. The grabbing hand, for instance, is suitable for jobs that resemble grasping or pulling, but it works just fine as an eraser, too. One exception is our old friend, the bee, whose significance is already defined in clear and narrow terms. Unless you're writing software for apiarists, it's confusing (and, hence, lousy GEM etiquette) to use the bee shape to signify anything other than "busy."

In addition to the pointer-changing routine (labeled CHANGE) the program demonstrates VDI routines which read the mouse button, make the pointer invisible, and force it back onto the screen. The routine labeled CLICK calls VDI routine 124, which can read the pointer's screen coordinates as well as monitor button activity. To read the pointer's *x* and *y* coordinates, add this line to the program:

```
305   print "x=";peek(ptsout),
       "y=";peek(ptsout+2)
```

The subroutines HIDE and SHOW call VDI routines that disable and enable the mouse pointer, respectively. If you don't hide the pointer before you change its shape, it may misbehave, depositing an unwanted ghost image in some cases. Watch out for such unexpected side effects whenever you call a GEM routine from BASIC. It's fun to manipulate GEM artifacts such as the pointer, but with that added power comes an extra measure of responsibility.

The BASIC Difference

Calling GEM routines from BASIC is significantly different from using them in a language like C or Pascal. Some system routines are downright antagonistic to BASIC, others are a waste of time, and others are redundant. The first difference arises because BASIC is itself a GEM application—a large, complicated program with its own ideas about what should be happening at any given time. Certain GEM routines shouldn't be used because they conflict with BASIC's own manipulation of the GEM environment.

The second category of routines includes those which do a job already performed by BASIC. For instance, since BASIC provides an output window, it's usually not necessary to open a virtual workstation or obtain a device handle before you call a system routine that draws on the screen. In the third category are routines that duplicate an existing BASIC command; why call a VDI routine to draw a circle, when CIRCLE is more convenient and achieves exactly the same result?

There's a fourth—fortunately, quite large—category of GEM routines: those which are both useful and usable from BASIC. In the months to come, we'll look at more of them.

```
100   fullw 2:clearw 2
110   for j=0 to 7 :rem Show al
      l 8 pointers
120   gosub HIDE:gosub CHANGE:g
      otoxy 1,1
130   print:read shape$:print s
      hape$
140   print "Click left button
      to continue..."
150   gosub SHOW:gosub CLICK
160   next j
170   gosub HIDE :rem Restore t
      he arrow
180   j=0:gosub CHANGE
190   closew 2:gosub SHOW
200   end
210   HIDE: poke contrl,123:rem
       Hide pointer
220   vdisys(0):return
230   SHOW: poke contrl,122 :re
      m Show pointer
240   vdisys(0):return
250   CHANGE: a#=gb :rem Key to
       Pandora's box
260   gintin=peek(a#+8) :rem Fr
      om me to AES
270   poke gintin,j :rem New mo
      use shape
280   gemsys(78):return
290   CLICK: poke contrl,124:re
      m Read mouse
300   vdisys(0)
310   if peek(intout)<>1 then C
      LICK
320   return
330   data Ye Olde Arrow,I-Beam
       Cursor
340   data Busy Bumblebee,"Poin
      ting Hand "
350   data Grabbing Hand,Skinny
       Crosshair
360   data Chubby Crosshair,Hol
      low Crosshair
```
©

Programming the TI

C. Regena

Game Programming

Many computer games are translations of games that already exist in some other form. The challenge in making such a conversion is to offer features that make you want to play the game on a computer instead of the usual way (with cards, dice, a board, or whatever). In the next two columns, we'll construct a game that has been popular under various names, but is usually called "Solitaire."

The original Solitaire game consists of several pegs arranged in a pattern of holes on a board. The center hole is left without a peg. Your goal is to get rid of pegs by jumping: One peg jumps over another into an adjacent hole, then the jumped peg is removed. You keep jumping and removing pegs until you can no longer jump. The optimum solution is to end up with one peg in the center hole. Actually, if you end up with one peg anywhere, you are an excellent player, and even two, three, or four remaining pegs would be a good score.

Why create this game on a computer? The main reason is that you'll often start to play the game, but find that some pegs are missing. You can't even set up the board without the right number of pegs. The computer will always set up the game without losing pegs, and can also check for impossible moves and thus prevent cheating. In a computerized version, we can also include a feature which would allow backing up and changing a move, or even replaying several moves. As a final enhancement, the program can keep track of every move in the game and print them out so you could prove to a friend that you really solved the puzzle.

I usually start game programming by designing the graphics. This playing board consists of yellow circles for the pegs and black circles for the holes. Lines 190–240 define graphic characters and colors, and lines 250–280 define strings for printing the board. The subroutine in lines 620–770 prints the starting board on the screen.

The next step is to move the pegs. CALL KEY is used for keyboard input. Use the arrow keys to move to the peg you want to move, then press ENTER. Now press an arrow key to show which direction to jump. The computer then needs to check to see whether you made a valid move.

Since the complete program is too long to include in a single column, I've split it into two separate portions. This month's listing includes enough of the program to draw the graphics and move the pegs, so you can play a complete game. However, not all of the features are included. Next month's column will explain more of the programming techniques and add the sections that let you back up to change a move, replay the game, or make a game printout.

If you to prefer to save typing time, you may obtain a copy of the complete program by sending a check for $3 together with a stamped, self-addressed mailer and a blank cassette or disk to:

C. Regena
P. O. Box 1502
Cedar City, Utah 84720

Be sure to specify the title, "Solitaire" for the TI-99/4A.

```
100 REM  SOLITAIRE
110 DIM G(12,12),M$(43)
120 CALL CLEAR
130 PRINT TAB(5);"** SOLITA
    IRE **"
140 PRINT ::"MOVE A PEG BY
    JUMPING OVER"
150 PRINT :"ANOTHER PEG TO
    AN EMPTY HOLE"
160 PRINT :"THEN REMOVE THE
     JUMPED PEG."
170 PRINT :"TRY TO END WITH
     ONLY ONE"
180 PRINT :"PEG IN THE CENT
    ER HOLE."
190 CALL CHAR(96,"0")
200 CALL CHAR(97,"0000183C3
    C18")
210 CALL CHAR(98,"00183C7E7
    E3C18")
220 CALL COLOR(9,11,7)
230 CALL CHAR(105,"00183C7E
    7E3C18")
240 CALL COLOR(10,2,7)
250 A$="````````"
260 B$="`a`a`a`"
270 C$="``````"&A$&"``````"
```

```
280 D$="`a`a`a"&B$&"a`a`a`"
290 FOR J=Ø TO 12
300 FOR K=Ø TO 12
310 READ G(J,K)
320 NEXT K
330 NEXT J
340 DATA 2,2,2,2,2,2,2,2,2,
    2,2,2,2
350 DATA 2,2,2,2,2,2,2,2,2,
    2,2,2,2
360 DATA 2,2,2,2,2,1,1,1,2,
    2,2,2,2
370 DATA 2,2,2,2,2,1,1,1,2,
    2,2,2,2
380 DATA 2,2,2,2,2,1,1,1,2,
    2,2,2,2
390 DATA 2,2,1,1,1,1,1,1,1,
    1,1,2,2
400 DATA 2,2,1,1,1,1,Ø,1,1,
    1,1,2,2
410 DATA 2,2,1,1,1,1,1,1,1,
    1,1,2,2
420 DATA 2,2,2,2,2,1,1,1,2,
    2,2,2,2
430 DATA 2,2,2,2,2,1,1,1,2,
    2,2,2,2
440 DATA 2,2,2,2,2,1,1,1,2,
    2,2,2,2
450 DATA 2,2,2,2,2,2,2,2,2,
    2,2,2,2
460 DATA 2,2,2,2,2,2,2,2,2,
    2,2,2,2
470 PRINT :::"PRESS <ENTER>
    ."
480 CALL KEY(Ø,K,S)
490 IF K<>13 THEN 480
500 CALL CLEAR
510 PRINT "USE THE ARROW KE
    YS THEN"
520 PRINT "<ENTER> TO SELEC
    T THE PEG,"
530 PRINT "THEN PRESS AN AR
    ROW KEY TO  MOVE."
540 PRINT ::"PRESS FCTN-8 T
    O REDO A PLAY."
550 PRINT ::"PRESS FCTN-5 T
    O SHOW ALL"
560 PRINT "MOVES FROM THE S
    TART."
570 PRINT ::"PRESS FCTN-P T
    O PRINT THE"
580 PRINT "SEQUENCE OF MOVE
    S."
590 PRINT :::"PRESS <ENTER>
     TO START NOW."
600 CALL KEY(Ø,K,S)
610 IF K=13 THEN 780 ELSE 6
    ØØ
620 CALL CLEAR
630 FOR T=1 TO 3
640 PRINT TAB(11);A$
650 PRINT TAB(11);B$
660 NEXT T
670 FOR T=1 TO 3
680 PRINT TAB(5);C$
690 PRINT TAB(5);D$
700 NEXT T
710 PRINT TAB(5);C$
720 FOR T=1 TO 3
730 PRINT TAB(11);B$
740 PRINT TAB(11);A$
750 NEXT T
760 CALL HCHAR(14,16,1Ø5)
770 RETURN
780 GOSUB 620
790 PRINT ::
800 R=6
810 C=4
820 ROW=R*2
830 COL=C*2+4
840 CALL GCHAR(ROW,COL,GG)
850 CALL KEY(Ø,K,S)
860 CALL HCHAR(ROW,COL,96)
870 CALL HCHAR(ROW,COL,GG)
880 IF S<1 THEN 850
890 IF K=13 THEN 1100
900 IF K<>69 THEN 950
910 IF R-1<2 THEN 850
920 IF G(R-1,C)=2 THEN 850
930 R=R-1
940 GOTO 820
950 IF K<>83 THEN 1ØØØ
960 IF C-1<2 THEN 850
970 IF G(R,C-1)=2 THEN 850
980 C=C-1
990 GOTO 830
1000 IF K<>68 THEN 1050
1010 IF C+1>1Ø THEN 850
1020 IF G(R,C+1)=2 THEN 850
1030 C=C+1
1040 GOTO 830
1050 IF K<>88 THEN 850
1060 IF R+1>1Ø THEN 850
1070 IF G(R+1,C)=2 THEN 850
1080 R=R+1
1090 GOTO 820
1100 CALL SOUND(5Ø,14ØØ,2)
1110 IF GG=105 THEN 850
1120 CALL KEY(Ø,K,S)
1130 CALL HCHAR(ROW,COL,98)
1140 CALL HCHAR(ROW,COL,97)
1150 IF S<1 THEN 1120
1160 IF K<>69 THEN 1240
1170 IF (G(R-2,C)<>Ø)+(G(R-
     1,C)<>1)THEN 153Ø
1180 G(R-1,C)=Ø
1190 CALL HCHAR(ROW-2,COL,1
     Ø5)
1200 CALL HCHAR(ROW,COL,1Ø5
     )
1210 G(R,C)=Ø
1220 R=R-2
1230 GOTO 1470
1240 IF K<>83 THEN 1320
1250 IF (G(R,C-2)<>Ø)+(G(R,
     C-1)<>1)THEN 153Ø
1260 G(R,C-1)=Ø
1270 CALL HCHAR(ROW,COL-2,1
     Ø5)
1280 CALL HCHAR(ROW,COL,1Ø5
     )
1290 G(R,C)=Ø
1300 C=C-2
1310 GOTO 1470
1320 IF K<>68 THEN 1400
1330 IF (G(R,C+2)<>Ø)+(G(R,
     C+1)<>1)THEN 153Ø
1340 G(R,C+1)=Ø
1350 CALL HCHAR(ROW,COL+2,1
     Ø5)
1360 CALL HCHAR(ROW,COL,1Ø5
     )
1370 G(R,C)=Ø
1380 C=C+2
1390 GOTO 1470
1400 IF K<>88 THEN 1100
1410 IF (G(R+2,C)<>Ø)+(G(R+
     1,C)<>1)THEN 153Ø
1420 G(R+1,C)=Ø
1430 CALL HCHAR(ROW+2,COL,1
     Ø5)
1440 CALL HCHAR(ROW,COL,1Ø5
     )
1450 G(R,C)=Ø
1460 R=R+2
1470 G(R,C)=1
1480 ROW=R*2
1490 COL=C*2+4
1500 CALL HCHAR(ROW,COL,97)
1510 CALL SOUND(5Ø,14ØØ,2)
1520 GOTO 840
1530 CALL SOUND(1ØØ,135,2)
1540 GOTO 850
```

©

INSIGHT: Atari

Bill Wilkinson

Five-Year Retrospective

This month marks my fifth anniversary writing "INSIGHT: Atari" for COMPUTE!. In the course of the last five years, I've covered a lot of different topics. Just for fun, I decided to look back through the last 60 issues of COMPUTE! and engage in some healthy self-criticism—listing the worst of Wilkinson as well as the best.

You may or may not agree with my assessments. But the point isn't simply to rate what's been done. After five years of writing about the same family of machines, it can be difficult to come up with a fresh topic every month. As you read these lists, let me know about some new topics you want me to cover, or some old topics that could stand further explanation or a fresh treatment. Not all of you have been reading COMPUTE! for a full five years, after all. And even long-time programmers can grow rusty in certain areas. This column is designed to serve you, the readers, so please provide some feedback in a card or letter addressed to:

Bill Wilkinson
P.O. Box 710352
San Jose, CA 95171-0352

The Brightest And Best

First, here's what I consider the best of "INSIGHT: Atari." Whether you agree will depend on your own viewpoint and needs. I have listed articles chronologically within broad categories.

- Getting more out of Atari BASIC: 9/81, 10/81, 12/81, 4/82, 5/82, 2/83, 1/84, 2/84, 3/84, 12/85, 3/86
- Calling I/O and GRAPHICS routines from assembly language: 11/81 through 2/82, 7/82 through 10/82, 8/85 through 10/85
- Assembly language techniques, with or without Atari BASIC: 12/81, 4/82, 10/82, 12/82, 7/83 through 9/83, 1/84, 12/84, 1/85, 3/85, 2/86, 4/86
- Converting BASIC programs to assembly language: 12/81, 2/82, 8/82 through 10/82, 5/84 through 7/84
- Atari BASIC internals: 1/82 through 7/82
- Bugs in Atari BASIC: 11/81, 5/85, 6/85
- Benchmarks: 9/82, 1/84, 11/84, 2/85, 3/85
- Playing music and sounds in background while a BASIC program runs: 3/82
- User definable function keys: 5/82
- Undocumented graphics mode: 10/83 and 11/83
- Using the extended memory of XL machines (with pictorial map): 12/83

Not So Memorable

Now for the less memorable columns. Some of my self-appointed projects have met with less than enthusiastic response. Perhaps the worst of these was "BAIT," a pseudo-BASIC interpreter written in Atari BASIC. The program was supposed to show you how language interpreters worked: It was so slow that you could literally watch the FOR-NEXT loops plod along. I prolonged the agony for four months (March, May, June, and August 1983).

Then I tried to rescue 1050 disk drive owners with an enhanced version of DOS 2.0S. It worked, but I doubt that more than a couple of dozen readers managed to get it installed properly. This series appeared May through September 1984. Less than four months later, we reworked DOS 2.0S for Atari to produce DOS 2.5. More time and energy down the drain.

My April Fool's columns have always received mixed reviews. This year, I got distracted and actually forgot to do a joke column. A couple of readers wrote me to compliment me on my restraint. Thanks, folks.

Some of the funniest installments of "INSIGHT: Atari" were unintentionally humorous, consisting of various predictions regarding future Atari products. I could have done better with a ouija board.

In addition to the obvious honkers, I've omitted from this list several columns which were relevant at the time they were written, but have since become outdated. One general regret is that I covered certain topics in less depth than now seems desirable. But that's a difficult factor to measure. When I invite you to explore a subject, do you ever sit down to research it further? If so, then I have succeeded. If not, perhaps the topic is inappropriate, or the treatment needs to be refined. Again, the more feedback you provide, the better I can meet your needs.

Truth Stranger Than Fiction

Since I just made fun of my precognitive powers, it's only fair to mention that one of my predictions is actually coming true. In July 1984, Jack Tramiel and company had just bought Atari. I wrote a column (published in October that same year) containing several predictions about what the "new" Atari would produce. On some points, I was correct: The 1450 died quickly, and the "Atari MAC" was already under development (it became what is now the ST).

Though it caused chuckles at the time, I also stated that Atari would continue to produce game machines and that they would soon come out with the already-designed 7800. As it happened, Atari sold over a million 2600 game machines in 1985. And, at the 1986 Summer Consumer Electronics Show, Atari announced that the 7800 will be available this autumn. Now, how would you like to know what's in store for 1988? ©

AmigaView

Charles Brannon, Program Editor

The Operating System

Amiga has released beta-test copies of version 1.2 of the operating system. These experimental versions are being distributed to software developers, but Amiga is encouraging informal distribution to help them get as much testing as possible. There will be a few more beta versions released, and we should see version 1.2 (which may actually be called version 2.0) out by Christmas.

However, it is also reported that Amiga is preparing to replace the WCS (Writeable Control Store, the area of RAM used to store the Kickstart portion of the operating system) with EEPROM (Electrically Erasable Programmable Read Only Memory), finally burning the operating system permanently into ROMs on the motherboard. This would have to be the final version, since replacing ROMs, if bugs are later discovered in the operating system, is not a trivial task.

The End Of WCS?

If Amiga replaced the WCS with ROM, we would lose the advantage of WCS: the ability to upgrade to a new (and even completely different) operating system at any time. On the other hand, there would be no need for a Kickstart disk, so booting up wouldn't take as long. 256K of ROM is cheaper than 256K of RAM, so this may be Amiga's primary consideration. But does Amiga plan to offer this ROM upgrade to current Amiga owners, or will we just use a Kickstart containing the equivalent of what gets burned into ROM?

Not everyone is clear on the hierarchy of the Amiga operating system, popularly referred to as Intuition. Although Intuition is fundamental, it's only part of the complete operating system (OS). There are actually several layers in the Amiga OS, which can be grouped into four major categories: Exec, Graphics, Intuition, and DOS.

Exec is the core of the operating system and controls every machine language program. Every task in the Amiga is part of a *task list*, and each task has a priority. Tasks with the most priority are allowed to run first. Whenever a task "goes to sleep" while waiting for something (keyboard or disk input, graphics, a response from another task, and so forth), the next highest priority task is allowed to run. However, no task is allowed to run longer than 64 milliseconds, the unit of time defined as a *quantum*. When a task's quantum is up, it is put to sleep to allow other lower-priority tasks to take their turn. Exec also contains subroutines for allocating and deallocating chunks of memory, and low-level input/output routines for accessing Amiga devices directly.

The Graphics library performs all the screen drawing functions such as line, rectangle, filled rectangle, and polygon drawing (and in version 1.2 includes functions for drawing hollow or filled circles and ovals). It contains powerful routines for animating graphic objects (bobs) and virtual sprites (vsprites), as well as providing direct access to the sprite hardware. In addition, the Graphics library allows programmers to modify the copper list, which controls the vertical aspect of the display. If you count the Layers library and Diskfont library as part of the Graphics library, the package also manages overlapping screen areas and multiple text fonts and styles.

Remarkable Flexibility

Intuition draws upon the resources of Exec and the Graphics library to create the high-level metaphors of windows, screens, menus, and gadgets. Intuition is large and complex, but it offers the programmer a remarkable level of flexibility. AmigaDOS uses Intuition for its CLI (Command Line Interface) and console windows, and Workbench relies heavily on Intuition to support its illusion of a desktop. Intuition is clearly the most visible part of the Amiga operating system (and probably the most important), but it cannot run on its own.

AmigaDOS is the topmost level of the operating system, the last part written, and was contracted from MetaComCo in England. Most Amiga applications are considered AmigaDOS *processes*, as opposed to Exec tasks. The Workbench is a layer above AmigaDOS, an application that creates a graphic world which performs many of the same functions as an AmigaDOS CLI without the cumbersome typing required by a command-driven DOS. AmigaDOS is much more than just a CLI, though. It includes the tools programmers need to read, write, and manage files and directories, rather than having to resort to direct track and sector access, as well as routines to load and execute programs as processes.

All these parts work in harmony (well, to be honest, with a few sour notes here and there) to orchestrate the complete Amiga system. You boot Kickstart, which loads in Exec, Intuition, and the Graphics library. You then insert a Workbench disk, which boots AmigaDOS and, finally, the Workbench. You open Workbench windows via Intuition and AmigaDOS, and execute applications, which have full access to all Amiga resources, even if many other programs are running at the same time. You can build your own unique working environment by choosing which programs you'd like to run together, and customize other options via Preferences. And when you add extra memory and peripherals, you have a symphony of exceeding range and power. ©

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a program to check your typing—"The Automatic Proofreader."

Programs for the IBM, TI-99/4A, and Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these Commodore and Atari characters in curly braces; *do not type the braces.* For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

Graphics characters entered with the Commodore logo key are enclosed in a special bracket: ⦃A⦄. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S̲. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S̲}, or ⦃8 Q⦄, you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse video

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <	◤	Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE	◀	Backspace
{DELETE}	ESC CTRL DELETE	◧	Delete character
{INSERT}	ESC CTRL INSERT	◨	Insert character
{DEL LINE}	ESC SHIFT DELETE	⬒	Delete line
{INS LINE}	ESC SHIFT INSERT	⬓	Insert line
{TAB}	ESC TAB	▶	TAB key
{CLR TAB}	ESC CTRL TAB	◧	Clear tab
{SET TAB}	ESC SHIFT TAB	◨	Set tab stop
{BELL}	ESC CTRL 2	◳	Ring buzzer
{ESC}	ESC ESC	⬑	ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:		See:
{CLR}	SHIFT	CLR/HOME	♥
{HOME}		CLR/HOME	S
{UP}	SHIFT	↑ CRSR ↓	●
{DOWN}		↑ CRSR ↓	Q
{LEFT}	SHIFT	← CRSR →	▌
{RIGHT}		← CRSR →	]
{RVS}	CTRL	9	R
{OFF}	CTRL	0	■
{BLK}	CTRL	1	■
{WHT}	CTRL	2	E
{RED}	CTRL	3	£
{CYN}	CTRL	4	◥
{PUR}	CTRL	5	▒
{GRN}	CTRL	6	↑
{BLU}	CTRL	7	←
{YEL}	CTRL	8	π

When You Read:	Press:		See:
⦃1⦄	COMMODORE	1	♣
⦃2⦄	COMMODORE	2	◜
⦃3⦄	COMMODORE	3	╳
⦃4⦄	COMMODORE	4	◘
⦃5⦄	COMMODORE	5	┴
⦃6⦄	COMMODORE	6	▌
⦃7⦄	COMMODORE	7	○
⦃8⦄	COMMODORE	8	▚
{ F1 }		f1	▬
{ F2 }	SHIFT	f1	◝
{ F3 }		f3	▬
{ F4 }	SHIFT	f3	◟
{ F5 }		f5	▌
{ F6 }	SHIFT	f5	◞
{ F7 }		f7	▐
{ F8 }	SHIFT	f7	■
←	←		←

key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as {SPACE}.

Amiga program listings contain only one special character, the left arrow (←) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN or move the cursor off the line to enter that line into memory. Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do *not* use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenable the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR(1536) to reenable it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a *checksum*.

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Program 1: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:REA
    D A:POKE I,A:CK=CK+A:N
    EXT I
120 IF CK<>19072 THEN ? "E
    rror in DATA Statement
    s.  Check Typing.":END
130 A=USR(1536)
140 ? :? "Automatic Proofr
    eader Now Activated."
150 END
160 DATA 104,160,0,185,26,
    3,201,69,240,7
170 DATA 200,200,192,34,20
    8,243,96,200,169,74
180 DATA 153,26,3,200,169,
    6,153,26,3,162
190 DATA 0,189,0,228,157,7
    4,6,232,224,16
200 DATA 208,245,169,93,14
    1,78,6,169,6,141
210 DATA 79,6,24,173,4,228
    ,105,1,141,95
220 DATA 6,173,5,228,105,0
    ,141,96,6,169
230 DATA 0,133,203,96,247,
    238,125,241,93,6
240 DATA 244,241,115,241,1
    24,241,76,205,238
250 DATA 0,0,0,0,0,32,62,2
    46,8,201
260 DATA 155,240,13,201,32
    ,240,7,72,24,101
270 DATA 203,133,203,104,4
    0,96,72,152,72,138
280 DATA 72,160,0,169,128,
    145,88,200,192,40
290 DATA 208,249,165,203,7
    4,74,74,74,24,105
300 DATA 161,160,3,145,88,
    165,203,41,15,24
310 DATA 105,161,200,145,8
    8,169,0,133,203,104
320 DATA 170,104,168,104,4
    0,96
```

Program 2: IBM Proofreader

By Charles Brannon, Program Editor

```
10 'Automatic Proofreader Vers
   ion 3.0 (Lines 205,206 adde
   d/190 deleted/470,490 chang
   ed from V2.0)
100 DIM L$(500),LNUM(500):COLO
    R 0,7,7:KEY OFF:CLS:MAX=0:
    LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,C
    HR$(4)+CHR$(70):ON KEY(15)
     GOSUB 640:KEY (15) ON:GOT
    O 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PR
    INT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT
    (LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POK
    E 1052,34:POKE 1054,0:POKE
     1055,79:POKE 1056,13:POKE
     1057,28:LINE INPUT L$:DEF
     SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$
    =MID$(L$,2):GOTO 170
```

```
1B0 IF VAL(LEFT$(L$,2))=Ø AND
    MID$(L$,3,1)=" " THEN L$=M
    ID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'no
     line number, therefore co
    mmand
205 BL=INSTR(L$," "):IF BL=Ø T
    HEN BL$=L$:GOTO 206 ELSE B
    L$=LEFT$(L$,BL-1)
206 LNUM=VAL(8L$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
210 IF TEXT$="" THEN GOSUB 540
    :IF LNUM=LNUM(P) THEN GOSU
    8 560:GDTD 150 ELSE 150
220 CKSUM=Ø:FOR I=1 TO LEN(L$)
    :CKSUM=(CKSUM+ASC(MID$(L$,
    I))*I) AND 255:NEXT:LOCATE
     Y,1:PRINT CHR$(65+CKSUM/1
    6)+CHR$(65+(CKSUM AND 15))
    +" "+L$
230 GOSU8 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT$:GOTO 150
    'replace line
240 GOSUB 5B0:GOTD 150 'insert
     the line
260 TEXT$="":FOR I=1 TO LEN(L$
    ):A=ASC(MID$(L$,I)):TEXT$=
    TEXT$+CHR$(A+32*(A>96 AND
    A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," ")
    :COMMAND$=TEXT$:ARG$="":IF
     DELIMITER THEN COMMAND$=L
    EFT$(TEXT$,DELIMITER-1):AR
    G$=MID$(TEXT$,DELIMITER+1)
     ELSE DELIMITER=INSTR(TEXT
    $,CHR$(34)):IF DELIMITER T
    HEN COMMAND$=LEFT$(TEXT$,D
    ELIMITER-1):ARG$=MID$(TEXT
    $,DELIMITER)
2B0 IF CDMMAND$<>"LIST" THEN 4
    10
290 DPEN "scrn:" FOR OUTPUT AS
     #1
300 IF ARG$="" THEN FIRST=Ø:P=
    MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-"):
    IF DELIMITER=Ø THEN LNUM=V
    AL(ARG$):GDSUB 540:FIRST=P
    :GOTD 340
320 FIRST=VAL(LEFT$(ARG$,DELIM
    ITER)):LAST=VAL(MID$(ARG$,
    DELIMITER+1))
330 LNUM=FIRST:GDSU8 540:FIRST
    =P:LNUM=LAST:GOSU8 540:IF
    P=Ø THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(S
    TR$(LNUM(X)),2)+" "
350 IF CKFLAG=Ø THEN A$="":GOT
    O 370
360 CKSUM=Ø:A$=N$+L$(X):FDR I=
    1 TO LEN(A$):CKSUM=(CKSUM+
    ASC(MID$(A$,I))*I) AND 255
    :NEXT:A$=CHR$(65+CKSUM/16)
    +CHR$(65+(CKSUM AND 15))+"
     "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLDSE #1:CKFLAG=Ø
400 GDTO 130
410 IF CDMMAND$="LLIST" THEN D
    PEN "lpt1:" FDR DUTPUT AS
    #1:GDTO 300
420 IF CDMMAND$="CHECK" THEN C
    KFLAG=1:GOTD 290
430 IF CDMMAND$<>"SAVE" THEN 4
    50
440 GOSU8 600:OPEN ARG$ FDR OU
    TPUT AS #1:ARG$="":GOTO 30
    Ø
450 IF COMMAND$<>"LOAD" THEN 4
    90
460 GOSUB 600:OPEN ARG$ FOR IN
    PUT AS #1:MAX=Ø:P=Ø
470 WHILE NDT EOF(1):LINE INPU
    T #1,L$:8L=INSTR(L$," "):B
    L$=LEFT$(L$,BL-1):LNUM(P)=
    VAL(BL$):L$(P)=MID$(L$,LEN
    (STR$(VAL(BL$)))+1):P=P+1:
    WEND
480 MAX=P:CLDSE #1:BOTO 130
490 IF COMMAND$="NEW" THEN INP
    UT "Erase program - Are yo
    u sure";L$:IF LEFT$(L$,1)=
    "y" OR LEFT$(L$,1)="Y" THE
    N MAX=Ø:LNUM(Ø)=65536!:GOT
    O 130:ELSE 130
500 IF COMMAND$="BASIC" THEN C
    OLOR 7,Ø,Ø:ON ERROR GDTO Ø
    :CLS:END
510 IF COMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 1
    30
540 P=Ø:WHILE LNUM>LNUM(P) AND
     P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:L
    NUM(X)=LNUM(X+1):L$(X)=L$(
    X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+1
     STEP -1:LNUM(X)=LNUM(X-1)
    :L$(X)=L$(X-1):NEXT:L$(P)=
    TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34)
     THEN 520 ELSE ARG$=MID$(A
    RB$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34)
     THEN ARG$=LEFT$(ARG$,LEN(
    ARG$)-1)
620 IF SEL=Ø AND INSTR(ARG$,".
    ")=Ø THEN ARG$=ARG$+".BAS"
630 SEL=Ø:RETURN
640 CLDSE #1:CKFLAG=Ø:PRINT"St
    opped.":RETURN 150
650 PRINT "Error #";ERR:RESUME
     150
```

Program 3: Commodore Proofreader

By Philip Nelson, Assistant Editor

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=Ø TO 166:READ 8YT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +8YT:NEXT
80 IF CHK<>20570 THEN PRINT "*
    ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*H8)
100 CHK=CHK+RF+LF+HF:POKE SA+L
   F,LB:POKE SA+HF,HB:NEXT
110 IF CHK<>22054 THEN PRINT "
   *ERROR* RELOAD PROGRAM AND
   {SPACE}CHECK FINAL LINE":EN
   D
120 POKE SA+149,PEEK(772):POKE
    SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
   14,22:POKE SA+18,23:POKESA+
   29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
   PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
   EEK(LO)+256*PEEK(HI))-1,Ø:N
   EW
160 DATA 120,169,73,141,4,3,16
   9,3,141,5,3
170 DATA 88,96,165,20,133,167,
   165,21,133,168,169
180 DATA Ø,141,Ø,255,162,31,18
   1,199,157,227,3
190 DATA 202,16,248,169,19,32,
   210,255,169,18,32
200 DATA 210,255,160,Ø,132,180
   ,132,176,136,230,180
210 DATA 200,185,Ø,2,240,46,20
   1,34,208,8,72
220 DATA 165,176,73,255,133,17
   6,104,72,201,32,208
230 DATA 7,165,176,208,3,104,2
   08,226,104,166,180
240 DATA 24,165,167,121,Ø,2,13
   3,167,165,168,105
250 DATA Ø,133,168,202,208,239
   ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
   211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
   85,211,3,32,210
280 DATA 255,162,31,189,227,3,
   149,199,202,16,248
290 DATA 169,146,32,210,255,76
   ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
   77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
   116,117,151,128,129,167,136
   ,137
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = Ø: FDR I = 768 TO 768 +
   68: READ A:C = C + A: PDKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   RDR IN PRDDFREADER DATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN PDKE 56,Ø: PDKE 57,3: CA
   LL 1002: GDTD 50
40 PRINT CHR$ (4);"IN#A$300"
S0 PDKE 34,Ø: HDME : POKE 34,1:
    VTA8 2: PRINT "PRDDFREADER
   INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 208,60,138,72,169,Ø
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,Ø,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,Ø,4,104,170
210 DATA 169,141,96
```

©

COMPUTE's Author Guide

Most of the following suggestions serve to improve the speed and accuracy of publication. COMPUTE! is primarily interested in new and timely articles on the Commodore 64/128, Atari, Apple, IBM PC/PCjr, Amiga, and Atari ST. We are much more concerned with the content of an article than with its style, but articles should be clear and well-explained.

The guidelines below will permit your good ideas and programs to be more easily edited and published:

1. The upper left corner of the first page should contain your name, address, telephone number, and the date of submission.

2. The following information should appear in the upper right corner of the first page. If your article is specifically directed to one make of computer, please state the brand name and, if applicable, the BASIC or ROM or DOS version(s) involved. In addition, *please indicate the memory requirements of programs.*

3. The underlined title of the article should start about 2/3 of the way down the first page.

4. Following pages should be typed normally, except that in the upper right corner there should be an abbreviation of the title, your last name, and the page number. For example: Memory Map/Smith/2.

5. All lines within the text of the article must be double- or triple-spaced. A one-inch margin should be left at the right, left, top, and bottom of each page. No words should be divided at the ends of lines. And please do not justify. Leave the lines ragged.

6. Standard typing paper should be used (no erasable, onionskin, or other thin paper) and typing should be on one side of the paper only (upper- and lowercase).

7. Sheets should be attached together with a paper clip. Staples should not be used.

8. If you are submitting more than one article, send each one in a separate mailer with its own tape or disk.

9. Short programs (under 20 lines) can easily be included within the text. Longer programs should be separate listings. *It is essential that we have a copy of the program, recorded twice, on a tape or disk.* If your article was written with a word processor, we also appreciate a copy of the text file on the tape or disk. Please use high-quality 10 or 30 minute tapes with the program recorded on both sides. The tape or disk should be labeled with the author's name, the title of the article, and, if applicable, the BASIC/ROM/DOS version(s). Atari tapes should specify whether they are to be LOADed or ENTERed. We prefer to receive Apple programs on disk rather than tape. Tapes are fairly sturdy, but disks need to be enclosed within plastic or cardboard mailers (available at photography, stationery, or computer supply stores).

10. A good general rule is to spell out the numbers zero through ten in your article and write higher numbers as numerals (1024). The exceptions to this are: Figure 5, Table 3, TAB(4), etc. Within ordinary text, however, the zero through ten should appear as words, not numbers. Also, symbols and abbreviations should not be used within text: use "and" (not &), "reference" (not ref.), "through" (not thru).

11. For greater clarity, use all capitals when referring to keys (RETURN, TAB, ESC, SHIFT), BASIC words (LIST, RND, GOTO), and three languages (BASIC, APL, PILOT). Headlines and subheads should, however, be initial caps only, and emphasized words are not capitalized. If you wish to emphasize, underline the word and it will be italicized during typesetting.

12. Articles can be of any length—from a single-line routine to a multi-issue series. The average article is about four to eight double-spaced, typed pages.

13. If you want to include photographs, they should be either 5×7 black and white glossies or color slides.

14. We do not consider articles which are submitted simultaneously to other publishers. If you wish to send an article to another magazine for consideration, please do not submit it to us.

15. COMPUTE! pays between $70 and $800 for published articles. In general, the rate reflects the length and quality of the article. Payment is made upon acceptance. Following submission (Editorial Department, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403) it will take from four to eight weeks for us to reply. If your work is accepted, you will be notified by a letter which will include a contract for you to sign and return. *Rejected manuscripts are returned to authors who enclose a self-addressed, stamped envelope.*

16. If your article is accepted and you have since made improvements to the program, please submit an entirely new tape or disk and a new copy of the article reflecting the update. We cannot easily make revisions to programs and articles. It is necessary that you send the revised version as if it were a new submission entirely, but be sure to indicate that your submission is a revised version by writing, "Revision" on the envelope and the article.

17. COMPUTE! does not accept unsolicited product reviews. If you are interested in serving on our panel of reviewers, contact the Review Coordinator for details.

CAPUTE!

64 Uncruncher

The first line was omitted from the MLX-format listing for this program in the August issue (p. 100). It should read as follows:

```
C000:AD 20 D0 8D 0A C6 A5 73 7D
```

Screen Machine II

When entering the program that accompanies Part 1 of this article in the July issue (p. 86), you'll encounter many lines for which the published "Automatic Proofreader" checksum will not match the one returned by the Proofreader even when the line is entered exactly as listed. The program in the July listing was generated by processing the commented listing from Part 2 of the article in the August issue (Program 1, p. 95) with the "REMover" program in that issue (Program 2, p. 99). REMover removes all comments, but in the case of comments at the end of program lines it leaves the space between the last BASIC statement and the apostrophe ('). This space affects the checksum calculated by our lister program, but cannot be typed when you enter the program (any spaces after the last character in a program line are ignored). Except for the Proofreader checksums, the July "Screen Machine II" program is correct as listed, so it should work if entered as listed without using the Proofreader. The checksums should all be correct in the commented (August) version.

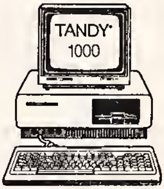
Apple ProDOS Catalog Sorter

The article with this utility program in the July issue (p. 96) states that the program can be modified for a 40-column video display simply by changing the PR#3 in line 260 to PR#0. Actually, several other changes are also required if you wish to display the sorted catalog on a 40-column screen: The HTAB statements should be removed from lines 340 and 780. The PRINT L2$: in line 460 should be changed to PRINT LEFT$(L2$,80 − 41 * (A$ <> "P")): and the PRINT DA$(I): in line 740 should be changed to PRINT LEFT$(DA$(I),80 − 41 * (A$ <> "P")):.

Also, the author has provided the following enhancement (this is not a correction). As published, the program sorts programs strictly by name. However, it's often useful to have programs sorted by type as well as by name, especially for directories on a hard disk. If you would like to modify the program to add this feature, change the assignment of the variable SK$(E) in line 680 to SK$(E) = MID$(L4$, 18, 3) + MID$(L4$, 2, 15).

Converting IBM ML to BASIC DATA

The article for this program erroneously states that this program will work on the PCjr. Cartridge BASIC for the PCjr does not support the SHELL command. (SHELL is included in Cartridge BASIC, but control does not return to BASIC after the command has executed.) Reader Wayne E. Robinson suggests a novel solution for PCjr owners: The PCjr normally uses Cartridge BASIC rather than either of the PC versions provided on the DOS disk, but it's not impossible to use the disk versions which properly support SHELL. When you type either BASIC or BASICA at a DOS A> prompt, DOS checks for the presence of Cartridge BASIC and displays an error message if no cartridge is found. You can trick the computer and use the disk versions of BASIC simply by changing their names. For example, you can use the ML-to-DATA program by using DOS to rename BASICA.COM as BASICB.COM, then typing BASICB (instead of BASICA) to start Advanced Disk BASIC, which can be used to run the program as listed. ©



Epyx Ships New Entertainment Packages

COMPUTE!'s coverage of the Summer Consumer Electronics Show (CES) in last month's issue inadvertently omitted significant new products from Epyx of Sunnyvale, California.

Epyx has introduced a variety of new entertainment programs for Apple, Commodore, Atari, and IBM computers.

Among the new releases are three bestselling packages recently converted for the Amiga and Atari ST computers: the classic *Temple of Apshai Trilogy,* three adventure role-playing games in one; *Rogue,* a 26-level graphic adventure game; and the popular *Winter Games,* featuring seven Winter Olympic contests. Epyx announced that many more of its most popular titles will appear in Amiga and ST versions later this year.

New titles include *The Movie Monster Game,* featuring the ever-popular Godzilla, an action game that lets the player take part in movie monster mayhem; *World Games,* a new series of Olympic contests in which players become international athletes and travel to eight different countries to compete in an event specific to each locale—for example, cliff-diving in Mexico or sumo wrestling in Japan; *Super Cycle,* an arcade-action motorcycle racing game that features a realistic first-person perspective; *Championship Wrestling,* a fast-action wrestling contest in which you choose your own wrestling personality and climb into the ring with a formidable opponent; and *World Karate Championship,* a graphically detailed karate-action program that features eight different compeition locations against increasingly difficult opponents. As with most earlier Epyx titles, these entertainment packages are available currently, or soon, for all major personal computer systems. Prices vary.

Epyx, Inc., 1043 Kiel Ct., Sunnyvale, CA 94089.

Circle Reader Service Number 170.

Color Printer Interfaces For Amiga, ST

The Okimate 20 now works with the Amiga and ST computers.

Okidata has announced that the Okimate 20, a color thermal transfer printer, can now be easily interfaced with the Amiga and Atari ST through its Plug 'N Print modules. The module is included in the $268 price for the printer, and contains everything the user needs to begin printing immediately: a data cable, black and color cartridge ribbons, and sample computer paper.

In addition to printing over 100 colors, the Okimate 20's 24-element printhead provides correspondence at 80 cps in draft mode and 40 cps in NLQ mode. Users can select from several different type fonts, including wide print, boldface, fine print, and italics. Underlining, superscript, and subscript are also standard features.

Okidata, 532 Fellowship Rd., Mt. Laurel, NJ 08054.

Circle Reader Service Number 171.

Electronic Greetings

Create and send electronic greetings—including animation and sound—with *Color Mail* from Hallmark Cards. This program lets you combine graphics, animation, music, sound, and personal messages to send greetings to other subscribers of CompuServe.

To use *Color Mail,* a subscriber develops a greeting offline and sends it through the electronic mail facility. The recipient transfers the greeting for offline viewing using his or her own *Color Mail* disk.

Color Mail can be ordered from CompuServe for $40. This includes CompuServe's VIDTEX communications program, 103 design elements, and illustrated user guides. A PalPak costs $60 and contains two disks, one for the sender and one for the recipient. There is a fee of 25 cents in addition to the connect time charge when using *Color Mail.* New groups of design elements can be ordered for $3.50 to $5.00.

Hallmark Color Mail, 2440 Pershing Rd., Ste G-40, Kansas City, MO 64108.

Circle Reader Service Number 172.

Database Manager For Commodore 128 And Amiga

Mid-Kansas Computers recently announced the release of Woodsoftware's *Flex File* for the Commodore 128 and Amiga, based on the earlier *Flex File* database manager for the 64 and PET computers.

On the Amiga version, all of the earlier command formats have been retained, and new features have been added that take advantage of the Amiga's power. These features include sophisticated virtual window entry editor with UNDO and CLEAR LINE functions; minimal mouse commands to speed data entry, editing, and processing; and storage of housekeeping data in machine memory to maximize file space. Two versions are included: An Amiga BASIC version that you can customize; and a machine language version for speed, multitasking with other programs, and more memory to handle extremely large and complex files. It retails for $79.95.

Flex File 128 is completely compatible with data disks created on earlier versions of *Flex File* and *Practifile* for other Commodore computers. Its command structure is identical to that of the older version, with a few enhancements. Up to 10,000 records can be created, with up to forty fields per record. 80-column FAST mode is supported, and HELP screens are available without disk access. It retails for $49.95.

Mid-Kansas Computers, 204 W. 6th, P.O. Box 506, Newton, KS 67114.

Circle Reader Service Number 174.

MECC Apple Educational Software

MECC has introduced two educational tools for Apple II series computers.

Quickflash! is a utility package that lets teachers create electronic flashcards. The program includes automatic recordkeeping, randomization of questions, control of mastery level, and printed progress reports.

Quickflash! can be adapted to various subject levels and includes diacritical marks and special characters for foreign language study. A printer option lets teachers print the questions and answers.

Students in grades six through nine can learn to write plays with *Show Time.* The students pick the cast from over 1000 possible combinations, build the sets, compose the music, and write the scripts using the integrated word processor, *MECC Writer.* With *Show Time,* students add stage directions, rehearse, edit the scripts, and finally watch the play. A support manual is included. Both *Quickflash!* and *Show Time* require an Apple II series computer with at least 64K. Contact MECC for prices.

MECC, 3490 Lexington Ave. N., St. Paul, MN 55126-8097.

Circle Reader Service Number 175.

Commodore 128 And IBM Compatibility

S.O.G.W.A.P. Software has introduced *The Big Blue Reader,* a software program that lets users transfer word processing and ASCII files generated on most IBM-compatible software to Commodore 128 DOS files, and vice versa.

Release 1.0 of *The Big Blue Reader* is priced at $29.95, plus $2 for shipping and handling (California residents add $1.95). *The Big Blue Reader* is self-booting. A full menu appears on the 80-column screen, while on the 40-column screen the program offers a main menu and submenus. Prompts take the user through the copying process, whether going from Commodore to IBM or IBM to Commodore.

The Big Blue Reader also offers the user the option of translating MS-DOS standard ASCII characters to Commodore ASCII characters—and vice versa—solving the problem of reversed capitals and lowercase letters.

S.O.G.W.A.P. Software, Inc., 611 Boccaccio Ave., Venice, CA 90291.

Circle Reader Service Number 176.

Pro Golf Simulator For Atari ST

Leader Board, for the Atari ST, is a realistic golf simulator that provides the player with a true perspective of the game. It features multiple 18-hole courses, 3-D animation, trees and sandtraps, and three levels of play. The program also provides for computerized scoring, a handicap system, and requires the player to make strategic decisions involving the choice of club, distance, and many other variables.

A joystick is required. The ST version of *Leader Board* retails for $39.95.

Access Software, Inc., 2561 S. 1560 W., Woods Cross, UT 84087.

Circle Reader Service Number 177.

RAM-Resident IBM Writing Tool

Micro Logic has released a RAM-resident productivity tool for the IBM-PC and compatibles. *Tornado Notes* lets you process random information using a system of parallel text processing. You can enter text into logical modules and then change, reorganize, and code the information as you wish. *Tornado Notes* has a flexible search capability and includes a pile-of-paper simulator, forms capability, note-joining function, two-keystrok duplication feature, and importing and exporting of both files and screens. There is a built-in editor as well as a helpful icon-based user interface.

Tornado Notes runs on the IBM-PC and compatibles with PC-DOS (MS-DOS) 2.0 or later and uses 50K of RAM, plus space for notes. It does not use bit graphics and supports most 80-character monochrome and color displays. The software is not copy-protected.

Tornado Notes costs $49.95, which includes a collection of reference notes and a 30-day money-back guarantee.

Micro Logic Corp., P.O. Box 174, 100 2nd St., Hackensack, NJ 07602.

Circle Reader Service Number 178.

Idea Processor For Amiga

Flow is an idea processor that takes full advantage of many of the Amiga's features, including multi-tasking, pull-down menus, windows, and the mouse.

The program's primary use is in organizing and arranging ideas in preparation for writing papers, articles, or books; or for presentations, planning, and decision-making. It can also be used to store and rapidly find important dates and appointments, or to save factual information in an orderly fashion. Suggested retail price is $99.95.

New Horizons Software, P.O. Box 43167, Austin, TX 78745.

Circle Reader Service Number 179.

BASIC Programming On The Apple

Thirty-five lessons in *Ace Programmer* cover the fundamentals of Apple BASIC programming on the Apple-II series computers. This new program from MindPlay instructs users, gives examples, and then offers students a chance to practice with 70 additional *playspace* assignments. The package includes recordkeeping, options to create additional playspace assignments, and a guidebook.

Ace Programmer is available on level I for grades 2 through 6 and level II for grades 7 through adult. Backup and lab packs are also available. Suggested retail price is $39.95.

MindPlay, Methods & Solutions, Inc., 82 Montvale Ave., Stoneham, MA 02180.

Circle Reader Service Number 180.

Hard Disk Drive For Commodore 64

The Data Chief is a hard disk drive system with floppy disk included for the Commodore 64, available in a 10-megabyte or 20-megabyte version. Produced by InConTrol, Inc., each system comes with a 170K floppy drive, a 135-watt power supply, a hard disk drive, and controller/driver cards, all housed in a metal case.

A second hard disk can be added without an additional driver card and, with an expansion kit that will be available this fall, three hard disks can be installed in the system. The Model HFD-60 is a 10-megabyte system

($895); the Model HFD-120 is a 20-megabyte system ($995).

InConTrol, Inc., 103 Baughman's Ln., Ste. 301, Frederick, MD 21701.
Circle Reader Service Number 181.

ST Versions Of Popular Text/Graphics Adventures

Spinnaker has announced that several titles in its popular Telarium series will now be available for Atari ST computers. The games include *Nine Princes Of Amber,* a game of negotiation, politics, and alliances in which you play a prince fighting for the throne of the one true perfect world (written by Roger Zelazny); *Amazon,* where as a special agent for a high-tech research firm you must travel to the dangerous, unexplored Amazon (written by Michael Crichton); and *Perry Mason: The Case Of The Mandarin Murder,* in which you play the role of world-famous criminal lawyer Perry Mason.

The ST versions of each program retail for $49.95.

Spinnaker Software, One Kendall Sq., Cambridge, MA 02139.
Circle Reader Service Number 182.

Commodore 16 And Plus/4 Programs

Two entertainment programs and a home finance package for the Commodore 16 and Plus/4 computers have been introduced by Robinson Software Associates.

Bounty Hunter is a text adventure set in the Old West; *Grave Robbers* is a graphic treasure-hunting adventure; and *Savings & Loan* is a home finance program that calculates principal, interest payments, amortization on loans, and various types of savings.

Each program sells for $9.95, plus $1.50 postage.

Robinson Software Associates (RSA), 50 South Valley Road B2, Paoli, PA 19301.
Circle Reader Service Number 183.

Star Micronics Printer

Star Micronics has introduced the NL-10, a 9-wire dot matrix desktop printer for professional, small office, and home use. The NL-10 prints high-speed draft quality at 120 cps and near letter quality at 30 cps. It offers eleven format and print functions, including three print pitch selections, type style, print mode, margin settings, and forward and reverse paper feed. The rear tractor feed has a quick tear feature plus an automatic feed. There is an optional automatic single and dual bin cut sheet feeder. Ribbon cartridges snap in easily.

The NL-10 has plug-in interface cartridges for the IBM PC and PC compatibles, Commodore 64/128, standard parallel computers, Apple computers, and an RS-232C serial interface cartridge.

The NL-10 dot matrix printer from Star Micronics is compatible with all major personal computers.

Suggested retail price for the NL-10 with one interface cartridge is $379. The base unit retails for $319 and each cartridge is priced at $60.

Star Micronics, Inc., 200 Park Ave., Ste. 3510, New York, NY 10166.
Circle Reader Service Number 184.

Inexpensive ST Software

Keypunch Software has introduced a line of inexpensive game, educational, and personal productivity programs for the Atari ST. Titles include *Trivia Master, The Gambler, Strategy Games, Cards Cards Cards, Mind Games, Personal Finance Pak, Executive Data Pak,* and *Finance I & II.*

Each program retails for $9.99. Amiga versions are planned for the fall of 1986.

Keypunch Software, 1221 Pioneer Bldg., St. Paul, MN 55101.
Circle Reader Service Number 185.

Macintosh Graphics

Dynamic Graphics has introduced *DeskTop Art* software for the Macintosh, a new line of programs that contains graphics selected and digitized from the company's library of more than 20,000 exclusive illustrations and photos. All images are based on original art, commissioned and purchased by Dynamic Graphics from leading illustrators for its international art services.

Each volume under the *DeskTop Art* name, categorized by subject and style, includes more than 300 illustrations stored on two disks as *MacPaint* documents. Also included in every package is a 24-page how-to guide, a pictorial index to the art, and suggested applications projects. The first two volumes are *Graphics & Symbols* ($66.95), a collection of high-contrast pictograms and symbols; and *Artfolio I* ($74.95), a miscellany of styles and subjects that includes people, familiar objects, and animals.

Dynamic Graphics, Inc., 6000 N. Forest Park Dr., P.O. Box 1901, Peoria, IL 61656-1901.
Circle Reader Service Number 186.

IBM Software From Buttonware

Buttonware has introduced several software packages for the IBM PC and compatibles.

PC-Dial is a communications package that features DOS access for commands or programs, complete support of DOS subdirectories, a built-in mini-editor for editing files online, support of user-defined scripts, smart keys that save up to 12 macros, a help screen, an automatic redial, communication at speeds from 75 bps up to 9600 bps, screen colors, and an on-screen timer. *PC-Dial* requires a serial communications port, a modem, DOS 2.0 or higher, 164K available RAM memory without the mini-editor and 220K of available RAM memory with the mini-editor.

PC-Style analyzes the readability of your writing by computing the percentage of long words, personal words, action verbs, words per sentence, and average syllables per word. This program works with any standard ASCII or *Wordstar* document.

PC-Tickle is a reminder program that helps you keep track of appointments, dates, and meetings. It also has an option that allows you to keep running totals of your checkbook balance, calorie consumption, and more.

PC-File III is a general purpose database manager program.

PC-File/R has more features than *PC-File III*, including relational database capabilities, integrated letter writing, and mail-merge capabilities.

A word processor, *PC-Type* can perform DOS functions and has keyboard macros as well as help panels to guide you through each process.

The graphics extension to *PC-File III* and *PC-File/R* is *PC-Graph*, which can plot a line graph of a database or a report created with the word processing programs.

PC-Dial, *PC-File III*, and *PC-Graph* each sell for $59.95. *PC-Style* and *PC-Tickle* each sell for $29.95 and *PC-File/R* costs $149.00.

ButtonWare, Inc., P.O. Box 5786,
Bellevue, WA 98006.

Circle Reader Service Number 187.

PBS Science Series Offers Free Software

Newton's Apple, the popular PBS science series, will introduce supplementary software to support this fall's series, thanks to a major grant from the Dupont Corporation.

The software series will consist of six Apple programs that deal with the scientific principles covered in the series. For example, as the host relates the laws of probability to the workings of a slot machine, a companion software program brings the lesson to the viewer through computer simulations of coin flipping, dice throwing, and slot machine playing. Additional software will be based on such program themes as mirrors, telescopes, and alcohol's effects on the body.

Newton's Apple software will be available at no cost on major online news and information services, local bulletin boards systems, user groups, and local board of education computer resource centers.

For further information, contact your local Apple user group or call a local FIDO-NET BBS.

Circle Reader Service Number 188.

Writing Aids For Apple II

I Can Write and *Be A Writer* introduce students to word processing as part of a book-building venture which encourages creative writing and helps teach basic grammar and writing skills. Both programs require the use of the Magic Slate, a Sunburst educational aid.

Challenges offered by *I Can Write*, designed for second graders, range from open-ended explorations of personal identity to changing a monster's description with new adjectives or commanding its actions with different verbs. Sudents can easily change or add to each exercise, then print out individual lessons to become part of their own personal writing record. In addition, they can create their own books of original stories, poems, letters, and drawings.

In *Be A Writer*, designed for third graders, students explore the narrative, descriptive, and explanatory styles of writing with imaginative characters like Ruby Robot and Giant George.

Both programs, available for Apple II computers, consist of 25 lessons each, and retail for $40.

Sunburst Communications, Inc., 39 Washington Ave., Pleasantville, NY 10570.

Circle Reader Service Number 189.

Statistical Baseball Game

SubLogic has introduced *Pure-Stat Baseball*, a statistical baseball simulation game originally being released for the Commodore 64, with later versions planned for the Apple II and IBM computers.

Pure-Stat Baseball contains every

major-league team from the 1985 season, along with eight classic teams from the past. The game, which is for one or two players, lets you trade team players, draft new players, or create your own teams. There are three stadiums to choose from on the game disk, or you can purchase an optional disk with every major league stadium in the U.S.

The emphasis throughout the game is on statistical realism. You select the team you want to manage, then pick the team you want to play against. Choose starting lineups, pitchers, make player substitutions, and call plays. Each player acts and moves individually on every play. The game maintains a complete statistical record as well.

The Commodore 64 version sells for $49.95. Versions for the Apple II and IBM PC computers will be released at a later date.

SubLogic Corp., 713 Edgebrook Dr., Champaign, IL 61820.

Circle Reader Service Number 190.

ST Cookbook On A Disk

Micro Cookbook, from FTL, consists of more than 150 recipes, and is an authoritative source of cooking tips and nutritional information designed to make you a better cook. It's a timesaving meal planner for organizing every menu detail. You pick the menu, and *Micro Cookbook* creates a shopping list of all the ingredients you'll need.

Available for the Atari ST, *Micro Cookbook* retails for $49.95.

FTL, 6160 Lusk Blvd., C206, San Diego, CA 92191.

Circle Reader Service Number 191.

Apple, IBM, Commodore PlayWriter Programs

Woodbury Software has announced the availability of two new programs in the company's PlayWriter series for the Apple II, Commodore 64, and IBM PC/PCjr computers. Each title in the series helps young authors write, edit, print, illustrate, and produce hardcover novels.

MYSTERY! and *Castles & Creatures,* the newest additions, are aimed at users age seven and above, including adults. In *MYSTERY!*, you write your detective novel by choosing and describing your sleuth, determining the method and motive of the murder, and creating your own cast of characters. In *Castles & Creatures,* you build your own adventure in a world of fantasy and imagination. Your environment is filled with dragons, knights, sorcerers, and royalty.

Each PlayWriter title is priced at $39.95 and includes a software story disk, color stickers, full-page illustrations, a hardcover book jacket, special paper, and easy to use instructions. Earlier PlayWriter titles include *Tales of Me* and *Adventures In Space.*

Woodbury Software, 127 White Oak Ln., CN 1001, Old Bridge, NJ 08857.

Circle Reader Service Number 192.

Commodore Music Software Guide

Commodore 64 & 128 Music Software Guide, by noted computer music consultant Lolita Walker-Gilkes, is a comprehensive music software guide that ranges from advice on how to use the Commodore for music to detailed explanations of individual software programs and their target audiences. The text presents descriptions, age groups, and prices, and breaks the information into sections on theory, eartraining, fingerings, composition, entertainment, and graphics. A separate section is devoted to MIDI (Musical Instrument Digital Interface), and appendices include vendor addresses, periodicals, and books that can further help users.

The guide sells for $11.95.

Unsinn Publications, P.O. Box 672, Drexel Hill, PA 19026.

Circle Reader Service Number 193.

Telecomputing Package

A new hardware and software package from Kinesis Corporation allows up to 23 simultaneous callers. *POPnet* lets users carry on private or open conversations with other users, take part in any of the two-player games, including chess, checkers, backgammon, and othello, or drop into one of the multiplayer games such as poker, liar, star trader, and house-o-fun. There are also mail and bulletin board areas.

POPnet is set up for operation as a business, complete with accounting software. Typical charges to a user is 75 cents an hour. Contact Kinesis Corp. for price.

Kinesis Corp., 3000 Citrus Circle, Suite 212, Walnut Creek, CA 94598.

Circle Reader Service Number 194.

Apple II, IBM Grammar Program

Grammar Gremlins, a comprehensive grammar program for elementary students, is the newest release from Davidson & Associates, for the Apple II+, IIe, and IIc at a suggested retail price of $49.95. An IBM version will be released in September.

Grammar Gremlins presents grammar rules with over 700 practice examples and sentences. The program covers abbreviations, subject/verb agreement, capitalization, contractions, parts of speech, plurals, possessives, punctuation, and sentence structure. Its features include an easy-to-use editor, animation, color, optional sound effects, record-keeping, and print-out capabilities.

Davidson & Associates, Inc., 3135 Kashiwa St., Torrance, CA 90505.

Circle Reader Service Number 195.

Commodore 64 Music

Free Spirit Software, publishers of the classical music disk, *Music of the Masters,* has announced a second classical music disk for the Commodore 64, *Music of the Masters, Vol. II.*

The program contains 40 compositions by composers such as Mozart, Bach, Beethoven, Brahms, and others. Instrument simulations include piano, harpsichord, violin, flute, guitar, and clarinet. Screen commentary on the composers is included.

Music of the Masters, Vol. II, has a price of $9.95. Both volumes may be purchased for $16.95. No shipping and handling charges.

Free Spirit Software, Inc., 5836 S. Mozart, Chicago, IL 60629.

Circle Reader Service Number 196.

Commodore Bulletin Board

Blue Board from SOTA Computing Systems is a bulletin board system for the Commodore 64 that supports over 200 online messages (of up to 1,023 characters), up to 220 users, and more than 25 sysop-definable sub-boards.

Written entirely in machine language, the system includes remote SYSOP access, a private sysop sub-board, and unlimited session connect time. *Blue Board* also includes *Scribbles,* which are mini sub-boards for messages of up to 80 characters (for opinion forums, voting, chess games, etc.). The system can be reconfigured by the sysop.

Blue Board requires a Commodore 64 or 128 with one disk drive (1541 or equivalent), and a 300-baud autoanswer modem (Commodore 1650 or equivalent). The suggested retail price is $69.95 (U.S. funds).

SOTA Computing Systems, Ltd., 213-1080 Broughton St., Vancouver, British Columbia, Canada V6G 2A8.

Circle Reader Service Number 197. ©

ATARI 130XE

ATARI 130XE Super Computer Package

130XE Computer	1027 Printer
1050 Disk Drive	Atariwriter +

Call for individual & super package price

ATARI PRINTER INTERFACES

Uprint A	54 95	MPP 1150	54.95

Supra 1000E Modem	**44.95**
Atari XM-301 Modem	**39.95**

ATARI 130-XE SUPER PRINTER PACKAGES

NX-10 Printer & U-Print A	299
Panasonic 1091 & U-Print A	299

Super Printer Packages have no extra shipping charges or credit card surcharges when shipped in Continental USA

ATARI 130XE SOFTWARE

BROOERBUNO

Print Shop	28 95
Karateka	20.95
Print Shop Graph. I, II, or III	19 95
Prinf Shop Comp.	27 95

INFOCOM

See Commodore 64 section for items and prices

ELECTRONIC ARTS

Archon II	24 95
Archon	19 95
Seven Cit. of Gold	24.95
Skyfox	24 95
Pinball Const.	19.95
One on One	24.95
Super Boulder Oash	19 95
Chessmaster 2000	27 95
Racing Destruction	24.95
Ultima III	34 95
T-down Football	23 95

MICROPROSE

Silent Service	23 95
Gunship	23 95
Accrojet	23 95
F-15 Strike Eagle	23 95
Kennedy Approach	23.95
Conflicf/Vietnam	27 95

SSI

See Commodore 64 section for items and prices

MISCELLANEOUS 130XE

Mind Pursuit	18 95
Never-Ending Story	18 95
Fooblitzky	27 95
Syncalc	32 95
Typesetter	24 95
Ultima III	34 95
Ulfima IV	41 95
Beachead II	27 95
Raid Over Moscow	23 95
Fight Night	19 95
Hardball	19 95
Flight Simulator II	34 95
Alternate Reality	24 95
Fleef System II	49.95
Page Designer	21.95
Megafont II	17 95
Rubber Stamp	21.95
Halley Project	23.95
Synfile	32 95
Beachead II	23.95
Music Studio	23.95
Wizard's Crown	27 95
Gettysburg	39.95
Island Caper	23 95
Basic XE	49.95
Action	47.95
MAC65XL	47 95
Paper Clip/Spell	39 95

ATARI ST

Atari 1040ST-RGB System	Call
Atari 1040ST-Monochrome System	Call
Atari 520ST-RGB System	Call
Atari 520ST-Monochrome System	Call
SF314DS/DD Disk Drive	209

We warranty all 520ST computers purchased from ComputAbility for ninety days.

MISCELLANEOUS ST

Gato	34 95
Kings Duest II	33 95
ST Talk	17.95
H&D Base	64 95
Typesetter ST	24 95
Word Invaders	24.95
Mean 18 Golt	34 95
Financial Cookbook	34 95
Brataccus	33 95
Michtron Utilities	39 95
Black Cauldron	27 95
Printmaster ST	24 95
PC Intercom	89.95
Final Word	94 95
Hex	27 95
Mudpies	27 95
Sundog	24 95
Flipside	27 95
Sottspoot	27 95
VIP Professional	119 95
Ultima II	39 95
Perry Mason	33.95
Degas	27 95
Farenheit 451	33 95
Personal Pascal	49 95
Art Gallery I	19 95
Compubridge	19 95
Major Motion	27 95
Pawn	29 95
DFT	34 95
Universe II	49 95
Macrodesk	20 95
Rubber Stamp	24 95
Kissed	34 95
Cards	27 95
Rouge	27 95
Mastertype	27.95
Easy Record	54 95
Swiftcalc ST	59 95
Data Manager ST	59 95
Little Comp People	34 95
Philon	Call
Wordwriter ST	59.95
Amazon	33 95
DOS Shell	27 95
Kissed	34 95
9 Princess/Amber	33 95
Hacker	29 95
Dragonworld	33.95
Treasure Island	27 95
Wizard of Oz	27.95
Borrowed Time	33.95
Mi-Term	34.95
Regent 2 w/Gem	64 95
Regent Spell	31 95
Regent Base	64 95
Goldrunner	27.95
Time Bandit	27.95
Zoomracks	49.95
Easy Draw	99.95
Mindshadow	33 95
Phantasie	27.95
Hippopotamus	Call
Supra Hard Drive	Call
Supra 1200 ST Modem	159
OMI 1200 ST Modem	159
PC Board Designer	Call
Infocom	See IBM
Apshai Trilogy	27.95
N-Vision	27 95
DB Man	69 95
Small Bus Pkg	84 95
Cornerman	34 95
ALT	34 95
M-Copy	54 95
Winter Games	27 95
Winnie The Pooh	19 95
Silent Service	27.95
Flight Simulator II	34 95
Music Studio	39 95
Phantasie	27 95
1 Megabyte Upgrade	175
Extended Warranty	Call
Computer Baseball	27 95
Chessmaster 2000	32.95
Strip Poker	27 95

AMIGA

Cell for Hardwere end Add-on Peripherals prices

AMIGA SOFTWARE

Hacker	29 95
Mindshadow	29.95
Mastertype	27 95
VIP Professional	139.95
Analyze	69 95
On-line	49 95
Artic Fox	29.95
Adventure Const	29.95
Deluxe Video	Cafl
Deluxe Paint	69 95
Infocom	See IBM
Monkey Business	19 95
Rouge	27 95
Apshai Trilogy	27 95
Delta Patrol	19.95
Aegis Draw	134 95
Little Comp People	34 95
Instent Music	34 95
Mean 18	34 95
Leader Board	27.95
Maxicom	39 95
Maxidesk	49.95
Maxiplan	119 95
One on One	29 95
7 Cities/Gold	29 95
Skytox	29 95
Marble Madness	29 95
Return/Atlantis	29 95
Archon	29 95
Aegis Animator	99 95
Deluxe Print	69 95
Music Studio	39 95
Borrowed Time	29 95
Financial Cookbook	34.95
Scribble	69 95
Halley's Project	29 95
Kid Talk	39.95
Speller Bee	39.95
Gizmo	34 95
DB Man	69.95

APPLE

SOFTWARE

BRODERBUND

Print Shop	39.95
Print Shop Graphics I, II, or III	17 95
Print Shop Comp	27 95
Karateka	21 95

ELECTRONIC ARTS

Bard s Tale	29 95
Auto-Duel	34 95
Skyfox	27 95
Lords of Conquest	27.95
Ultima IV	39 95
Moebius	39 95
Amnesia	Call

EPYX

Winter Games	24 95
Summer Games II	24 95

INFOCOM

See IBM section for items and prices.

MICROPROSE

See Atari 130XE section for items and prices.

SIR-TECH

Wizardry/Diam	23.95
Wizardry/Legacy	27 95
Wizardry/Proving	33 95

SSI

See Commodore 64 section for items & prices.

APPLE MISCELLANEOUS

Gamemaker	34 95
Hardball	24 95
Sundog	27.95
Newsroom	39 95
Clip Art Vol I	20 95
Clip Art Vol II	27 95
Gato	27 95
Kung Fu Master	25 95
Karate Champ	25 95
Strip Poker	23 95
Math Blaster	34 95
Fight Night	24 95
Phantasie II	27.95
Rambo	27 95
Amer Challenge	27 95
Crossword Magic	34 95

IBM PC

Corona PC-400 Compatible	Call
Corona Portable PC Compatible	Call

IBM PC SOFTWARE

BROOERBUND

Print Shop	39 95
Print Shop Graph I	27 95
Bank Street Writer	49 95
Ancient Art of War	29 95

MICROPROSE

F-15 Strike Eagle	23 95
Kennedy Approach	27 95
Acrojet	27 95
Silent Service	27 95

IBM MISCELLANEOUS

Mean 18	34 95
Gato	27 95
Wizardry	39.95
Strip Poker	27.95
Jet	34 95
Newsroom	39 95
Superbowl Sunday	24 95
Home Pak	34 95
Alter Ego	34 95
Hacker	29 95
Borrowed Time	27 95
Isgur Porttolio	164 95
Typing Tutor	34 95

INFOCOM

Deadline	34 95
Starcross	34 95
Zork I	27 95
Zork II or III	29 95
Witness	27 95
Suspended	34 95
Planetfall	27 95
Sorcerer	29 95
Seastalker	27 95
Cutthroats	27 95
Hitchiker	27 95
Suspect	29 95
Wishbringer	27 95
Infidel	29 95
Enchanter	27.95
Spellbreaker	34 95
Mind Forever	34 95
Ballyhoo	27.95

SIERRA

King's Duest	34.95
King's Quest II	34 95
Ultima II	39.95

COMMODORE

COMMODORE 128

C-128 Computer	Call
1571 Disk Drive	Call
1902 Monitor	Call
1670 Modem	Call
1350 Mouse	42 95
1750 512K Expander	Call

COMMODORE 128 SOFTWARE

Swiftcalc w/Side.	49 95
Wordwriter + Spelf	49 95
Data Manager II	49 95
Fleef System III	54.95
Mach 128	34 95
Superbase 128	69.95
Pocket Writer 128	32 95
Pocket Planner 128	32 95
Superscript 128	59 95
Perfect Writer	49.95
Sylvia Porter-128	49 95
Perfect Filer	49.95
Viziwrite 128	Call
Vizistar 128	Call
Pockef Filer 128	32 95
Partner 128	39.95

COMMODORE 64

NEW 64C Computer	Call
C-64 Compufer	Call
1541 Disk Drive	Calt
1660 Modem	49 95
1670 Modem	Call
1802 Monitor	Call

COMMODORE 64/128

SUPER PRINTER PKGS.

NX-10 & Xetec Supergraphic	299
Panasonic 1091 & Xetec Supergraphic	309
Legend 1080 & Xetec Supergraphic	269

Super Printer Packeges hava no added shipping or charge card surcherges when shipped in Continentel USA

While Supplies Last

B.I. 80 Column Card	**59.95**
XETEC Super Graphic	**69.95**
Cardco G-Wiz	**54.95**

GENERAL HARDWARE

star micronics·inc

THE POWER BEHIND THE PRINTED WORD.

NX-10	**Call**
SG-10	**Call**
SG-15	369
SD-10	339
SD-15	449
SR-10	489
SR-15	Call
NL-10	Call
NL-10C	Call

PRINTERS

Panasonic 1091	239
Legend 808	169
Legend 1080	209
Powertype	309
Juki	Call
Epson	Call
Panasonic 1080	215

PRINTER BUFFERS

Microfazer	From 169
U-Buff 16K	79 95
U-Buff 64K	99 95

MOOEMS

Volksmodem 1200	189
Prometheus 1200	299
Maxwell 1200	Call
Maxwell 2400	Call
Prometheus 2400	Call

MONITORS

NAP Amber	89.95
Commodore 1802	Call
Goldstar 13" Color	129
Teknika RGB	Calt
Magnovox RBG	Call
Commodore 1902	Cafl

COMMODORE 64 SOFTWARE

ACCESS

Mach V-Cart	21 95
Leader Board	27.95
Tenth Frane	27 95

SSI

Battle/Antietam	32 95
Panzer Grenider	24 95
USAAF	37.95
Kampgruppe	37 95
Broadside	24 95
Carrier Force	37.95
Comp. Ambush	37.95
Field of Fire	24 95
Gemstone Warrior	21 95
Imp Galactum	24 95
Computer Baseball	24 95
Comp. Ouarterback	24 95
Wizards Crown	27 95
Gettysburg	39 95
Phantasie	24 95
Mech Brigade	39 95

SSI (NO ATARI)

Rings of Zilfin	27.95
Phantasie II	27 95
Battle Group	39 95
Roadwar 2000	27 95

MICROPROSE

See Atari 130XE section for items and prices.

ABACUS SOFTWARE

Call for Items end Prices

We stock hundreds of programs for the Apple, Atari, C-64 and IBM. If you don't see it here, don't hesitate to call.

INFOCOM

Zork I	24.95
Zork II, or III	27.95
Deadline	29 95
Slarcross	29.95
Witness	24.95
Planetfall	24 95
Hitchiker	24.95
Enchanfer	24.95
Cufthroafs	24.95
Sorcerer	24 95
Spellbreaker	29.95
Ballyhoo	27.95

ELECTRONIC ARTS

Adv. Construction	29 95
Mail Order Monster	24 95
Racing Destruction	24 95
Ultima IV	41.95
Bard's Tale	27 95
Lords of Conquesf	27 95
Chessmaster 2000	27.95
Ultimafe Wizard	24 95
Super Boulderdash	19 95
Mind Mirror	24 95
Auto Duel	34 95
Ogre	27 95

See Atari 130XE section for rest of items & prices.

EPYX

Winter Games	24 95
Apshai Trilogy	24 95
Fasf Load-Cart	24 95
Vorpal Ulility	22 95
Multiplan	44 95
Movie Monster	24 95
World Karate	19 95
Super Cycle	24 95
Champ. Wresfling	24 95

MISCELLANEOUS COMMOOORE 64

Freeze Frame	34.95
Print Shop	28.95
Cal-Kit	34.95
Superbase 64	47.95
Karateka	20.95
Hacker	20.95
Gamemaker	27.95
Ultima II	37.95
Karate Champ	25.95
Sticks of Death	20 95
Kung Fu Master	25.95
Paper Cfip/Spell	54.95
Consultant	41.95
Spy vs. Spy Vol. II	23 95
Internaf'l Hockey	19 95
Prt Shop Compan.	27.95
Prt Shop Graphics	19.95
Gato	20.95
Jet	29 95
Fleef System II	44 95
Printmasfer	24.95
Newsroom	34.95
Fontmaster II	34 95
Sports Lib. Vol. I	19 95
Geos	39 95
Hidden Assefs	Call
Adv. Music System	54 95
Spitfire 40	23.95
Fight Night	19 95
Hardball	19 95
2 on 2 Basketball	23 95
Murder/Mississippi	23.95

— EST. 1982 —

ComputAbility™

P.O. Box 17882, Mifwaukee, WI 53217
ORDER LINES OPEN
Mon-Fri, 11 a.m. - 7 p.m. CST
Sat. 12 p.m. - 5 p.m. CST

NO SURCHARGE FOR MASTERCARD & VISA

To Order Call Toll Free

800-558-0003

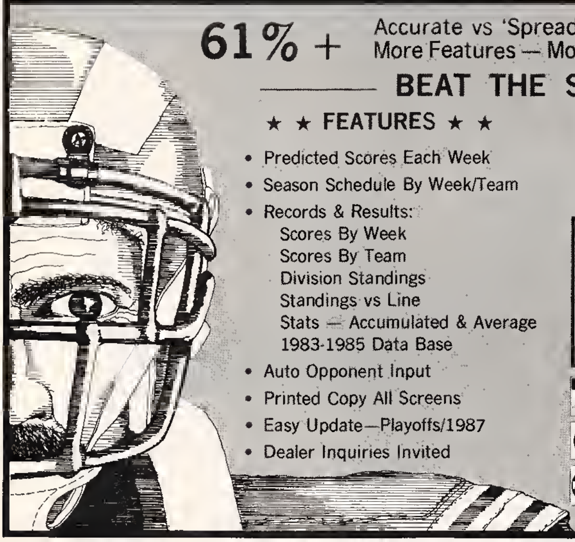
For Technical Info., Order Inquiries, or for Wisc. Orders

414-351-2007

ORDERING INFORMATION: Please **specify sysfem.** For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. **C.O.D. charges are $3.00 In** Confinental U.S.A. include $3.00 for software orders, 4% shipping for hardware, minimum $4 00 Masfer Card and Visa orders please include card #, expirafion date and signature. WI residents please include 5% sales fax. HI, AK, FPO, APO, Puerfo Rico and Canadian orders, please add 5% shipping, **minimum $5.00.** All other **foreign orders add 15% shipping, minimum $10.00.** All orders shipped outside fhe Confinental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed fhe minimum amounf, you will be charged fhe addifional amount to gef your package fo you quickly and safely. All goods are new and include factory warranfy. Due fo our low prices all sales are final. **All defective returns must have a return authorization number.** Please call (414) 351-2007 to obfain an R.A.# or your return will nof be accepfed. Priced and availabilify subjecf to change wifhouf nofice.

PIK'EM 86

★ ★ ★ ★ A Complete Pro Football Prediction Program For The 1986 NFL SEASON



61% + Accurate vs 'Spread' Since 1981

More Features — More Information In 1986 To . . . .

BEAT THE SPREAD

★ ★ FEATURES ★ ★

- Predicted Scores Each Week
- Season Schedule By Week/Team
- Records & Results:
  - Scores By Week
  - Scores By Team
  - Division Standings
  - Standings vs Line
  - Stats — Accumulated & Average
  - 1983-1985 Data Base
- Auto Opponent Input
- Printed Copy All Screens
- Easy Update—Playoffs/1987
- Dealer Inquiries Invited

54^{95} INCLUDES DISK, DOCUMENTATION SHIPPING & HANDLING

- Apple II — IIc — IIe
- Commodore 64 — 128
- IBM PC, Tandy & Compatibles
- TRS-80 MIII/IV

Stats Needed To Run Program
Available In Local Newspapers
Or
We Will Furnish Stats By Mail Or Modem
All 20 Weeks — Season Price . . . 40^{00}

Program Comes Updated Thru Current
Week Of Season . . . No Extra Charge

ORDERS SHIPPED IN 2 DAYS
PHONE LINES OPEN 24 HRS.

VISA C.O.D. MasterCard

TOLL FREE — 800-722-2277
TEXAS COLL. — 214-586-8212

MARATHON SOFTWARE DEPT. C
P. O. BOX 1349
JACKSONVILLE, TEXAS 75766

TOLL FREE & COLLECT FOR ORDERS ONLY

THE LOWEST PRICES THE BEST SERVICE

ELECTRONIC ONE*

PHONE LINES OPEN 10-6 E.S.T. M-F

CALL (614) 864-9994 • P.O. Box 13428 • COLUMBUS, OHIO 43213

COMMODORE HARDWARE

C128 COMPUTER	249 99
C64 COMPUTER	139.99
1541 COMPUTER	179.99
1571 DISK DRIVE	229.99
1702 MONITOR	179.99
1902A RGB MONITOR	269.99
MPS. 1000 PRINTER	239.99
1350 MOUSE	39 99
1200 BAUO MODEM	149 99

PRINTERS

STAR 10X	169.99
STAR SG 10C	229 99
STAR NX 10	239 99
PANASONIC 1080	199.99
PANASONIC 1091	229.99
EPSON LX80	229.99
LX80 TRACTOR	24.99
SEIKOSHA	179 99

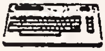
MISC. HARDWARE

G-WIZ INTERFACE	46.99
CARDCO-G INTERFACE	37.99
TYMAC INTERFACE	49.99
P.P.I. INTERFACE	29.99
XETEC JR.	36.99
XETEC SR. (8k BUFF)	56.99
TOTAL COMM. MODEM	29.99

MONITORS

14" THOMPSON COLOR	139.99
13" SAKATA COLOR	139.99
12" SAMUNG GREEN	49.99

commodore



DISKS

SONY S/S D/D	7.99
BASF S/S D/D	7.99
PERCISION D/D D/D	8.99
DISK NOTCHER	2.99
FLIP & FILE (SO)	6.99

JOYSTICKS

THE BOSS	11.99
THE BAT	16.99
KRAFT	7.99
SPECTRO VIDEO 1	4.99
SPECTRO VIDEO 2	4.99

SUPER SUMMER SOFTWARE SALE

JUKE BOX	2.99
STORY MACHINE	2.99
ALPHABET ZOO	2.99
UP FOR GRABS	2.99
DELTA DRAWING	2.99
AOVENTURE CREATOR	3.99
COSMIC LIFE	3 99
LOGIC LEVELS	3.99
FACE MAKER	3.99
SWIFTAX (1983)	3.99
SNAKE MAN	3 99
DANCE FANTASY	4.99
SEA HORSE	4.99
DUCKS AHOY	4.98
WEBSTER WORD GAME	4.99
LINKING LOGIC	4.99
TYPE ATTACK	4.99
BIG BIRD	4.99
SERPENTINE	4.99
MISSING LINKS	4 99
AZTEC	4.99

GORTEK MICRO CHIPS	99¢
JUICE	99¢
THE FACTORY	99¢
ATTACK MUTANT CAMELS	99¢
BRAVO	1.99
KID GRID	1 99
TURTLE TOYLAND	1.99
ALF COVER CAVES	2 99
TRAINS	2 99
SNOOPER TROOPS	3 99
HEY DIDDLE DIDDLE	3.99
CONGO BONGO	3 99
POOR MAN	3.98
TROLL'S TALE	3 99
SPACE RESCUE	3.99
JUNO FIRST	3 99
LEARNING W/LEEPER	3 99
JAW BREAKER II	3.99
OAVID MIDNIGHT MAGIC	3 99
BUSINESS SYSTEMS	4.99
PUZZLE PANIA	4.99

ALL QUANTITIES ARE LIMITED
SALE PRICES END SEPTEMBER 30
OR WHILE SUPPLIES LAST.

Q-BERT	4.99
BEAM RIOER	4.99
PITFALL	4.99
ROBOTS OF DAWN	4.99
HERO	4.98
HES GAMES	4.99
FROGGER	4.99
SLIOES, HOME FILE	4.99
MR. ROBOT	4.99
DIG OUG	4.99
CENTIPEDE	4.99
ROBOTRON	4.99
CHOPLIFTER	6.98
MIRAGE WORD PROCESSOR	7.99
ACCOUNTS PAYABLE	7.99
SIMONS BASIC	7.99
GHOSTBUSTERS	9.99
READING	9.99
BARRONS S.A.T.	19.99

VIC 20 SOFTWARE $1.^{99}$ EACH

- OEMON ATTACK
- FAST EDDIE
- ATLANTIS
- POLE POSITION
- CENTIPEDE
- GALAXIAN
- DONKEY KONG
- BATTLE ZONE
- DEFENDER
- ROBOTRON
- MOON PATROL
- MS. PAC MAN
- KRAZY ATTACKS
- JUNGLE HUNT
- TURMOIL

HOW TO ORDER: CASHIER CHECK, MONEY ORDER, MASTERCARD* OR VISA* (ADD 4% FOR CHARGE CARDS) . . NO PERSONAL CHECKS . . . NO C.O.D.'s . . . SHIPPED U.P.S. . . . ALL PRICES SUBJECT TO CHANGE WITHOUT NOTICE.
SHIPPING: ADD $3.00 ON ALL ORDERS UNDER $100.00 . . . ADD $5.00 ON ALL ORDERS OVER $100.00. ACTUAL FREIGHT CHARGED ON MULTIPLE ORDERS.
INTERNATIONAL: ACTUAL FREIGHT CHARGED ON ALL ORDERS OUTSIDE THE CONTINENTAL UNITEO STATES INCLUDING A.P.O.
POLICIES: NO RETURNS WITHOUT A RETURN AUTHORIZATION . . NO RETURNS UNLESS DEFECTIVE. ALL DEFECTIVES WILL BE EXCHANGED . . . NO EXCEPTIONS.

ELECTRONIC ONE
CALL: **(614) 864-9994**

WRITE: P.O. Box 13428
Columbus, Ohio 43213

ABBY'S

DISCOUNT SOFTWARE

COMMODORE AMIGA

256 K. Color Amiga System	CALL
256 K. Amiga	CALL
A1010 3½" Drive	CALL
A1050 256 K. Expansion	CALL

ATARI ST

520 ST Color System	CALL
520 ST Monochrome System	CALL
SF 314 Drive	CALL
SF 354 Drive	CALL

ATARI 8-BIT

800XL Computer	$ 69
130 XE Computer	$129
1050 Drive	$129
1027 Printer	$ 99
1020 Printer	$ 24
XM 301 Modem	$ 39

COMMODORE 8-BIT

064C Computer	$179
D1541C Drive	$199
C-128 Computer	CALL
VIC-1571 Drive	CALL
VIC-1581 Drive	CALL
512K RAM Expansion	$179

CALL ON THESE AND OTHER PRODUCTS
WE CARRY A COMPLETE LINE OF SOFTWARE AND HARDWARE INCLUDING PRINTERS

TOP HITS

ATARI SOFTWARE ST

Home Planetarium	$ 24
Joust	$ 24
CP/M Emulator	$ 39
2 Key Accounting	CALL
Little Computer People	$ 39
Championship Golf	CALL
Degas	$ 29
Music Studio	$ 37
Leader Board Golf	CALL
Financial Cookbook	$ 29
Fortran	$129
Jet	CALL
Flight Simulator II	$ 39
Pascal	$129
"C" Compiler	$129
Cobol	$259
Print Master	$ 29
Art Gallery	$ 24

ATARI SOFTWARE XL/XE

Battle of Antietam	$30
USAAF	$35
M.U.L.E.	$17
Movie Maker	$39
One On One	$24
Seven Cities of Gold	$24
Silent Service	$27
F-15 Strike Eagle	$25
Music Studio	$27
Printshop Companion	$21
Temple of Apshai Trilogy	$26
Flight Simulator II	$39
Scenery Disk (For Above)	$16
Karateka	$24
Nam	$29
Super Bowl Sunday	$22
Atari Basic (R)	$ 5
Print Shop	$35
Graphic Library I, II & III	ea. $18

COMMODORE SOFTWARE AMIGA

Artic Fox	$30
Sky Fox	$29
Deluxe Paint	$65
Deluxe Print	$65
Aegis Draw	CALL
Aegis Animator	$99
Music Studio	$39
One On One	$29
On Line	$49
Analyze	$69
Amiga Assembler	CALL
BBS	$65
Lattice "C"	CALL
Amiga Pascal	CALL
Scribble	$65

COMMODORE SOFTWARE 64/128

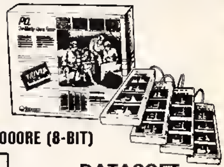
Mastertronics Title As Low As	$7.95
All Abacus Title	CALL
Football	$29
Printship	$35
Printshop Companion	$28
Leader Board Golf	$28
Tournament Disk (For Above)	$14
Print Master	$27
Newsroom	$39
Fontmaster II	$31
Fast Hack'em	$31
Snapshot	$39
Tenth Frame	CALL
Merlin 64/128	CALL
Bard's Tale	$28
Lords of Conquest	$29
Sky Fox	$27
USAAF	$35
Gunship	$25
Silent Service	$25

ATARI POWER PACKS

REPLACEMENT TRANSFORMER FOR 800/400, 810, 1050, 1200XL, 1020 . . . . $4.95

EARLY BIRD CHRISTMAS SPECIALS

SUNCOM
P.Q. PARTY QUIZ
ONLY
$14^95



ATARI OR COMMODORE (8-BIT)

RIBBONS

Atari 1025 (Black)	$2.50
Atari 1025 (Color)	$3.50
Panasonic 1091 (Black)	$8.50
Okimate 10 (Color)	$6.00
Star Gemini 10 (Black)	$2.50
Star Gemini 10 (Color)	$3.50
Star Powertype	$6.00
Epson MX/RX/FX-80	$6.00
Commodore 1525 (Black)	$7.50
Commodore 1526 (Black)	$7.00
Commodore MPS 801	$7.50
Commodore MPS 802	$7.00
Commodore MPS 803	$9.00

BULK DISKS

100 Quality DSDD 5¼"	ea. $ .69
100 Quality SSDD 5¼"	ea. $ .59
Less Than 100 DSDD 5¼"	ea. $ .79
Less Than 100 SSDD 5¼"	ea. $ .69
SONY DSDD 3½"	ea. $2.05

DATASOFT

Commodore 64/128

O'Riley Mine (D/C)	$5.95
Dallas Quest (D)	$5.95
Mancopter (D)	$5.95

ATARI XL/XE

Pooyan (D/C)	$5.95
Moon Shuttle (D/C)	$5.95

ATARI, INC.

ATARI XL/XE

Juggle's Rainbow (C)	$2.99
Juggle's House (C)	$2.99
States & Capitals (C)	$1.99
My First Alphabet (D)	$2.99

TIMEWORKS

COMMODORE 64/128

Money Manager (D/C)	$5.95
Electronic Checkbook (D/C)	$5.95
Data Manager (D/C)	$5.95

ATARISOFT

APPLE

Defender (D)	$2.99
Dig Dug (D)	$2.99
Robotron (D)	$2.99
Pac Man (D)	$2.99
Centipede (D)	$2.99
Donkey Kong (D)	$2.99

ALL TITLES ON DISK UNLESS MARKED (R) FOR ROM CARTRIDGE

WE CHECK FOR STOLEN VISA & MASTERCARD

Order Line
1-800-282-0333
ORDERS ONLY! Sales Personnel Do Not Have Information on Previous Orders or Product Specs.

M-F 10 a.m.-8 p.m.; Sat. 10-3 ET

37 S. Broad Street, Fairborn, OH 45324

Customer Service & Ohio Residents
1-513-879-9699

AD #CP-096

VISA

SOFTWARE ONLY — Prepaid orders over $50 receive free shipping via UPS in continental U.S. Please add $2 orders under $50. HARDWARE and all orders requiring shipment via U.S. Post Office are subject to additional freight charges. Add $5 for COD orders. VISA/MasterCard orders add $2 service charge. Ohio residents add 5.5% sales tax. Personal checks require a three-week waiting period. No waiting when paid by credit card, certified check or money order. All items subject to availability and price change. PLEASE CITE AD NUMBER WHEN ORDERING. ALL RETURNS MUST BE ACCOMPANIED BY AN AUTHORIZATION NUMBER.

MasterCard

$5 TALKING DISK

OVER 100 WORDS in vocabularies that you can put into your own programs! No extra hardware required. Sample programs include:

- Talking four-function calculator — choose English, Spanish, or German.
- Talking keyboard — letters and punctuation in English.
- Demonstration of voice editing.

The $5 Talking Disk is available for Commodore 64, 128, Atari 800, 800XL, 130XE, and Apple II+ (64K), IIe, and IIc.

If you want to make your own vocabularies, in any language or accent, then you must have a VOICE MASTER for processing speech input. VOICE MASTER lets you do much more: YOU CAN RECOGNIZE SPOKEN COMMANDS and WRITE MUSIC AS YOU HUM! And affordable too — under $90 including the headset and microphone.

Send $5 for the talking disk. Prepaid orders only. Includes postage. ($7 outside USA, Canada, and Mexico.) Information on VOICE MASTER will accompany your disk. Or you can call or write for VOICE MASTER information separately. Please specify computer make and model with your order.

COVOX INC.
675 Conger St., Dept. C!
Eugene, OR 97402
Telephone (503) 342-1271

FACTORY AUTHORIZED

COMMODORE REPAIR CENTER

1-800-772-7289
IN ILLINOIS (312) 879-2888

C64 Repair (cpu only)	42.95*	AMIGA Repair	CALL
1541 Repair	79.95*	C64 Power Supply	34.95
1541 Alignment only	29.95*	CBM & AMIGA PARTS	
C128 Repair (cpu only)	79.95*	Call (312) 879-2350	

*Includes parts, labor & UPS return shipping. APO/FPO or Air Freight add $10.00

CALL BEFORE SHIPPING, VISA, MASTERCARD OR MONEY ORDER
24-48 Hrs. Turnaround (Subject to parts availability)
Diagnosis fee of $25.00 for any unit altered or with no defects.

SAVE YOURSELF SOME MONEY

Due to our vast experience in repairing 64s we have found that the power supply is a major cause of board failures. By inserting our device between the power supply and computer over 63% of our customer's repairs would have been avoided. For only $19.95 our C64 Over Voltage Sensor will continuously monitor your power supply output. If an overvoltage occurs the Over Voltage Sensor will prevent damage to your 64.

64 OVER VOLTAGE SENSOR $19.95

TEKTONICS PLUS, INC.
150 HOUSTON STREET
BATAVIA, IL 60510

CLIP AND SAVE

Commodore Compatible and only. . . $139.00

FSD-1 5¼" Disk Drive

Directly replaces the Commodore 1541 disk drive.

Check these Features

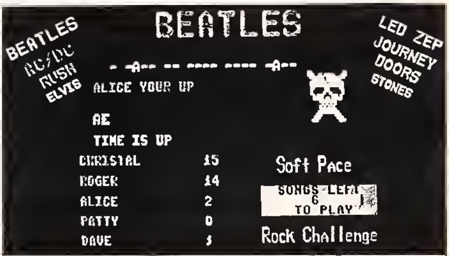
- Runs all C-64 Software — Including heavy copyright protected softwere
- Full 6 month warranty — your assurance of quality
- Heavy duty construction — to run whisper quiet and smooth year after yeer
- Built in 24 watts solid state power supply — runs cool and efficient always
- Vented metal chessis — eliminetes most redio-frequency interference
- Dual serial ports with cheining option for expandability
- External device number switches — for use as second disk drive
- Save and rsplace cepability
- Positive lever locking system — to eliminate "Pop out" problem
- Stenderd 5¼" SSSD Diekette, 175K storage capecity, 310 RPM running speed, 90 Sec. format time, 1200 BPS data trensfer — seme as the 1541
- Compatible with C-128 in C-64 mode

To Order Call Toll Free 1-800-356-5178

Visa & MasterCerd welcome. Credit Card orders shipped in 24 hrs. Allow $8.00 shipping and hendling. Send mail order with peyment to:

ECI

Emerald Component International Dept. N
541 Willamette Street
Eugene, OR 97401
Tel. 503-683-1154



```
                BEATLES

 - -A-- -- ---- ---- -A--
ALICE YOUR UP

 AE
 TIME IS UP
CHRISTAL        15      Soft Pace
ROGER           14      SONGS LEFT
ALICE           2           6
PATTY           0        TO PLAY
DAVE            3       Rock Challenge
```

Greatest party game ever ! Hundreds of songs in one of the most challenging and fun games out for the Commodore. No two games ever alike ! Up to 5 players at a time.

With Rock Challenge Group Editor you can add your favorite group's titles and save them to your own data disk. Later when the group comes out with more songs just add them to your data disk.

Example: You choose to play the Beatles with 35 songs to play. The program chooses one of the over 100 songs and places as a series of dashes. Player 1 guesses a letter in the song, if there he wins a point and continues his turn till he guesses wrong or solves the song for a 5 point bonus.

FAST MACHINE CODE PROGRAM

Only $14.95 + $ 1.50 Postage

Send check or money order to:
SoftPace Software Co
P.O. BOX 788
Brea, CA. 92622-0788

VISA

Visa or Mastercard call:
(800) 225-6207
In California call:
(714) 524-0343

MasterCard

Classified

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.) Inquire about display rates.

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Ad will appear in next available issue after receipt.

Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

SOFTWARE

FREE SOFTWARE for C64, C128, IBM, & CPM.
Send SA5E for info (specify computer) to:
PUBLIC DOMAIN U5ER5 GROUP
PO Box 1442-A1, Orange Park, FL 32067

DISCOUNT SOFTWARE: Amiga/Apple/Atari/C64-128/IBM PC-PCjr/TR5-80/Timex/Sinclair. Free Catalog: WMJ DATA 5Y5TEMS,
4 Butterfly Dr., Hauppauge, NY 11788

COMMODORE: TRY BEFORE YOU BUY.
Best selling games, utilities, educational, classics and new releases. 100's of titles. Visa/MC. Free brochure. RENT-A-DI5C, Frederick Bldg. #345, Huntington, WV 25701 (304) 529-3232

BRIDGE GAME PROGRAM $39.95 demo disk $5. 1 to 4 players for IBM, Apple, TI99, TR580, C64, 128, +4, 16, VIC. Authors John & Lynda Allan, Box 313, Azilda, Ontario, Canada P0M 1B0

ATARI 8 BIT: FULL FUNCTION BU5INE55 DBM5. Any drive(s)/upgrades. GL/AR/AP/Inv/Mail/W/P. 9000 records/disk. MICROMOD, 1635-A Holden, Orlando, FL 32809 (305)857-6014

$$ WIN WITH THOROUGHBRED HARNE55 & Greyhound handicapping software...$29.95, enhanced...$49.95. For most computers. Free info. Software Exchange, Box 5382 CP, W. Bloomfield, MI 48033. Call: (313) 626-7208

FREE APPLE SOFTWARE
Over 1000 Public Domain Programs on 50 diskettes. $5 each plus $1 shipping per order. 5end $1 for catalog (refundable with order.)
C&H ENTERPRISES
PO Box 29243, Memphis, TN 38127

Update & Analyze your income and expenses with INCOME ANALYSIS. Menus, hardcopies, totals, averages, percentages, summaries, fast-edit, sound & more. A MU5T for part-time employees! Atari 48K Disk only $12.95.
DATA LOGIC, POB 52193, Livonia, MI 48152

THE PICKS-86. PRO FOOTBALL PICKING GAME! Compete against your friends. Computer keeps track of stats. Send only $12.
Epic Software, Box 1153, West Hanover, MA 02339. Apple II family. 1 disk drive+printer. 5end SA5E for more information.

PRO FOOTBALL ANALYST. Beat the spread! Only 5 min./wk & local newspaper is all it takes to find hidden overlays in the betting line. Winning season guaranteed! IBM/C64/TI99-$34.95 (+ $4.55 s/h) ORDER NOW! 1-800-341-1950 Ext. 77. Mail Orders: RIDGE, 170 Broadway #201-C, NYC, NY 10038. Catalog.

LOTTO PICKER. Improve your chances for those Million Dollar Jackpots! Picks LOTTO, WIN-4 and Daily Numbers. All USA & CAN. games incl'd. Expandable! IBM/C64/TI99-$29.95. Order Now! 1-800-341-1950 Ext. 77. Mail Orders: Ridge, 170 Broadway #201-C, NY, NY 10038. Catalog.

PRINT YOUR OWN BUSINESS STATIONERY
Use your dot-matrix printer and this Business Papers Kit to make personalized invoices, letterheads, business cards, memos, etc. from plain paper. Try before you buy. 5end $7.50 to: Honor-System Software, Box 338, Medinah, IL 60157. Specify Apple II or C64.

SELL YOUR PROGRAMS to Software Publishers. 5oftware for the Apple, Atari, Commodore, IBM, TI, TR5-80, Zenith, Osborne, Kaypro, others. Directory $5.95. I.W., Box 40581, Pas., CA 91104

TI-99/4A QUALITY 5OFTWARE for Business, Home and Entertainment ** Bonus 5oftware Offer!!** Send for FREE catalog to MICRO-BIZ HAWAII, BOX 1108, PEARL CITY, HI 96782

HARDWARE

Clone Kits, Modems, Hard Drive Kits, Disk Drives, Diskettes, Printers, Memory & ICs Distributor Pricing to End Users and Dealers. Free 5hipping. For Catalog Call 1-800-833-2600, In Ohio (513) 531-8866

MISCELLANEOUS

IBM PCjr REPORT: THE NATIONAL NEWSLETTER. PCjr-specific articles, reviews, Public Domain from across the nation. $18/yr.
PCjr CLUB, POB 95067, 5chaumburg, IL 60195

100% ERROR FREE

DISKETTES

MFR. LIFETIME WARRANTY

With Hub Rings, Write Protect Tabs, Tyvec Envelopes, User ID Labels · In Factory/Sealed Poly Packs

(YOU GET EVERYTHING BUT THE BOX)
PRICES ARE PER DISK

5¼"			3½"	
SSDD	DSDD	DSHD96TPI	SS	DS
.47	.52	1.69	1.39	1.60

SUPER DISCOUNTS ON ORDERS OF 500 OR MORE

IN STOCK ITEMS SHIPPED IN 24 HOURS

Minimum order: $25.00. **Shipping and Handling:** $4.00 per 100 diskettes Continental U.S.A APO/PFO, add $8.00 per 100 diskettes. Foreign order, please call. Reduced shipping charge for larger quantities. C.O.D. add $4.00 Cash/certified check MI residents add 4% sales tax. Prices subject to change without notice. HOURS: 8:30 AM - 6:00 PM Eastern Time

COD

Precision Data Products
P.O. Box 8367, Grand Rapids, MI 49518
(616) 452-3457 • Michigan 1-800-632-2468
Outside Michigan 1-800-258-0028

DISK SALE!

PREMIUM QUALITY! LIFETIME WARRANTY!

Guaranteed 100% error free, money-back satisfaction guarantee! Made by top USA makers, not "seconds". We buy millions from the factory. We can't print the maker, but you'll recognize them as among the highest quality, premium diskettes made! Order BULK (K-pak) for no sleeve or box, BOXED IN TENS (D-pak) or BOXED IN 25's (J-pak) with label, sleeve, box and tabs. Order D-pak, add 2¢ per disk to boxed prices.

FROM 33¢

	BULK					BOXED				
Prices are per disk	100	500	1K	5K	10K	25	100	500	1K	5K
For the Apple II, II+, IIe, IIc #UBA	45c	42c	39c	36c	33c	58c	53c	50c	42c	44c
SS/DD for single side computers #UD1	50c	47c	44c	41c	38c	63c	58c	55c	52c	49c
DS/DD for IBM-PC and compatibles #UD2	55c	52c	49c	46c	43c	68c	63c	60c	57c	54c
DS/Hi Density for IBM-AT #UAT	1.75	1.72	1.68	1.65	1.60	1.90	1.83	1.79	1.76	1.73
3½" single side for Mac #UD3	1.49	1.44	1.41	1.39	1.37	1.59	1.49	1.44	1.41	1.39
3½" double side for Mac & Amiga #UD8	1.95	1.91	1.88	1.84	1.79	1.99	1.95	1.91	1.88	1.84
5¼" SLEEVES 24# white wove #SLV	5c	4.5c	4c	3.8c	3.3c	SPECIFY SUFFIX K J D WHEN ORDERING DISKS!				
5¼" SLEEVES white Tyvek #TYV	7c	6c	5.5c	5.2c	5c					

MORE GREAT PRICES!

LIBRARY DISK STORAGE CASES, clear plastic #F10 99¢ea, 10 for $9.50
FLIP-TOP DISK FILES, holds 70 disks, smoked plastic #F70 $8.88
UNI NOTCH DISKETTE NOTCHER, access 2nd side of disk #NN1 $7.77
COMPUTER MAILING LABELS, old 5 lines x 3.5" #LAB $19 box of 5000
HEAD CLEANER KIT, 1 cleaner disk & solvent, 15 uses #HCK $4.95
DB25 GENDER CHANGERS, all pins wired, choose M/M (#MMGC) or F/F (FFGC) $7.77
RIBBONS FOR OKIDATA 80/81/82/82A/83/83A/92/93 #PR180 $2.15ea 12/$23.65

HOW TO ORDER: Pay by MC-Visa-Amex-COD, or send check with order. Min order $20, add $3 for UPS shipping +$2 if COD (no extra charge for credit cards). We can ship open account to schools, min school purchase order $100, FOB Unitech. All orders must include daytime phone and STREET address. MONEY BACK 30-DAY SATISFACTION GUARANTEE! SEND FOR FREE CATALOG LISTING 100'S OF COMPUTER BARGAINS!

UNITECH
20-U HURLEY STREET
CAMBRIDGE, MA 02141-2176
IN MASS (617)"UNITECH"

(800)343-0472

Advertisers Index

Reader Service Number/Advertiser		Page
102	Abacus Software	27
103	Abacus Software	29
104	Abby's Discount Software	125
105	Acorn of Indiana	128
106	Alpha Systems	19
107	Artificial Intelligence Research Group	12B
	Batteries Included	11
108	Bridge Publications	47
	C.O.M.B. Direct Marketing Corp.	44
109	CompuServe	7
110	ComputAbility	123
111	Computer Mail Order	14,15
	Covox Company	126
	Crown Custom Covers	85
112	Davidson & Associates, Inc.	49
113	Digital Solutions, Inc.	IFC
114	Electronic One	124
115	Emerald Component International	126
116	Firebird Licensees, Inc.	23
117	Firebird Licensees, Inc.	51
118	Free Spirit Software, Inc.	12B
	Great Western	117
119	Jason Ranheim	50
	Jesse Jones Industries110	
	Lyco Computer	24,25
121	Marathon Software	124

Reader Service Number/Advertiser		Page
122	Micro Computer Services	121
123	Micro Marketing	83
124	MicroProse	9
	NRI Schools	97
125	Origin Systems Inc.	BC
126	Precision Data Products	127
127	Protecto	35,36,37
128	Silicon Express	53
129	SoftPace Software Co.	126
130	Springboard	4
131	subLOGIC Corporation	1
	Tektonics Plus, Inc.	126
132	Thompson Consumer Products	IBC
133	Time Life Books	17
134	Unitech	127

COMPUTE! Books' Atari ST Collection	13
COMPUTE! Classifieds	127
COMPUTE! Disk Subscription	48,101
COMPUTE! First Book of Commodore 128 and Electronic Computer Projects	55
COMPUTE! Subscription	33
COMPUTE!'s Apple Applications Special	64
COMPUTE!'s Atari ST Disk & Magazine Contest	2,3
COMPUTE!'s First, Second, and Third Book of Apple	39

WE TOP APPLE AND COMMODORE BEAUTIFULLY

Thomson offers a whole new world of graphics capabilities for your Commodore or Apple IIC or IIE computers. And Thomson offers you more than the competition.

Thomson monitors offer these advantages:

- Compatible with IBM,® Apple,® Commodore,® Atari™ and others
- The choice of monochrome or color monitors with TV-grade to high resolution text and graphics
- Monochrome text-editing mode for color monitors
- Alternate use as cable or VCR monitor
- Broad range of the features you need at the prices you want

Because of its versatility and compatibility, you can still use your Thomson monitor if you switch computers; you'll never outgrow the capabilities of a Thomson monitor.

Who is Thomson? Thomson is a six billion dollar multi-national company. Unlike our competition, we design and manufacture our own monitors; so the quality you get is constant and superb.

Your Apple or Commodore computer is a great body. Choose a Thomson monitor, and give it a beautiful face.

For the name of the Thomson dealer nearest you, call 1-800-325-0464. In California call 1-213-568-1002. (Monday-Friday, 9 a.m. to 5:00 p.m. PST)

THOMSON

A SIGHT FOR SORE EYES.™

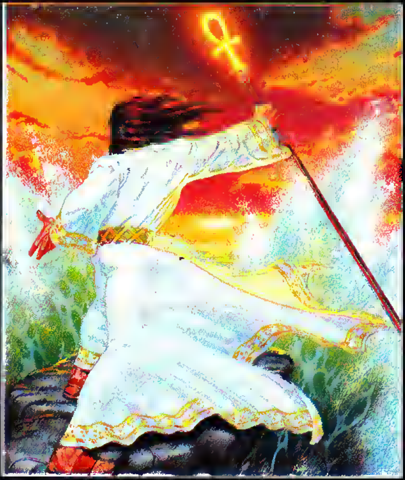
© 1986 Thomson Consumer Products Corporation 5731 W. Slauson Avenue, Suite 111, Culver City, CA 90230

Thomson is a trademark of Thomson S.A. Apple is a registered trademark of Apple Computer, Inc. Commodore is a registered trademark of Commodore Electronics Limited
Atari is a trademark of Atari, Inc. IBM is a registered trademark of International Business Machines Corp

From Origin comes the long-awaited sequel to the award-winning Ultima™ III

Ultima IV

Quest of the Avatar



A state-of-the-art fantasy role-playing game of unprecedented magnitude by Lord British™.

Prepare yourself for a grand adventure: Ultima™ IV, sixteen times larger than Ultima III, is a milestone in computer gaming—one that challenges your physical and mental skills while testing the true fabric of your character.

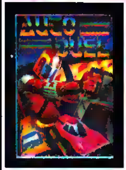
Enter Britannia, kingdom of Lord British. Journey through terrain of infinite proportions, conversing with characters on hundreds of topics. Unravel the mysteries of a superior magic system. At each turn beware of daemons, dragons and long-dead wizards haunting the most tranquil of places. Encounters with parties of mixed enemy types test your strategic abilities. Shrewd use of terrain can lead to victory against seemingly impossible odds.

Survive this multi-quest fantasy, then begin the final conflict, your quest of the Avatar. The ultimate challenge—the self—awaits....

ORIGIN SYSTEMS INC. 340 HARVEY ROAD, MANCHESTER, NH 03103 (603) 644-3360

ULTIMA™ III sends you on an incredible fantasy role-playing journey through monster-plagued Sosaria in search of the elusive Exodus.

MOEBIUS™ takes you through the elemental planes of a colorful Oriental world of fantasy and adventure in search of the Orb of Celestial Harmony.



AUTODUEL™ is a futuristic, fast-paced strategy role-playing game where the right of way goes to the biggest guns.

OGRE™ is a strategy game fought on the nuclear battlefield of tomorrow as an inhuman juggernaut Cybertank battles conventional forces.

Ultima and Lord British are trademarks of Richard Garriott/Moebius is a trademark of Greg Malone/AutoDuel and Ogre are trademarks of Steve Jackson/Apple is a trademark of Apple Computer Inc/Previous Ultimas are not needed to enjoy Ultima IV

Authors wanted. Call us today.